(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－187082
（Ｐ２００２－１８７０８２Ａ）
(43)公開日　平成14年７月２日(2002.7.2)

(51)Int.Cl.⁷	識別記号	Ｆ Ｉ		テーマコード（参考）
B25J 13/00		B25J 13/00	Ｚ	3C007
5/00		5/00	Ｃ	
G06N 3/00	550	G06N 3/00	550 Ｅ	

審査請求　未請求　請求項の数33　ＯＬ　（全25頁）

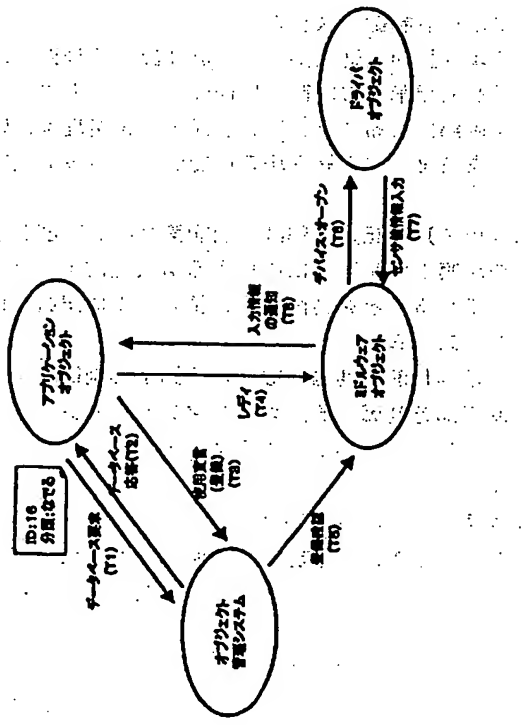
(21)出願番号　特願2001－313865(Ｐ2001－313865)
(22)出願日　平成13年10月11日(2001.10.11)
(31)優先権主張番号　特願2000－310033(P2000－310033)
(32)優先日　平成12年10月11日(2000.10.11)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　坂本　隆之
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　井上　真
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　100101801
弁理士　山田　英治　（外２名）

最終頁に続く

(54)【発明の名称】ロボット制御システム及びロボット制御方法

(57)【要約】
【課題】　ハードウェア依存のミドルウェア層と、ハードウェア非依存のアプリケーション層との組み合わせを動的に変更して多関節型ロボットを制御する。
【解決手段】　ロボットのハードウェア構成に依存するミドルウェア層と、ハードウェア構成に依存しないアプリケーション層の間に、意味的に動作を行うためのインターフェースとデータベースを用意することによって、ロボット上に導入するミドルウェアとアプリケーションの組み合わせを変更しても、常に正常な動作を保証することができる。また、アプリケーションは、ミドルウェアを介して適当な入力データを取得したり、適切なコマンドを発行することができる。



1

【特許請求の範囲】

【請求項１】複数のハードウェア構成要素の組み合わせからなるロボットの動作を制御するロボット制御システムであって、

ロボットのハードウェア構成情報に依存しない処理を行う第１の制御部とロボットのハードウェア構成情報に依存する処理を行う第２の制御部と前記第１及び第２の制御部間の通信を行う通信部と、を具備することを特徴とするロボット制御システム。

【請求項２】前記第１の制御部は、ロボットの構成や動作を抽象化したモデルによってロボットの行動シーケンスを決定するアプリケーション・ソフトウェアを実行することにより実現されることを特徴とする請求項１に記載のロボット制御システム。

【請求項３】前記第１の制御部は、ロボットの感情をモデル化した感情モデルと、本能をモデル化した本能モデルと、外部事象とロボットがとる行動との因果関係を逐次記憶していく学習モデルと、行動パターンをモデル化した行動モデルのうち少なくとも１つを備えることを特徴とする請求項１に記載のロボット制御システム。

【請求項４】前記第２の制御部は、ロボットの機体上における基本的な機能を提供するミドルウェア・ソフトウェアを実行することにより実現されることを特徴とする請求項１に記載のロボット制御システム。

【請求項５】前記第２の制御部は、ロボットのハードウェアから検出される入力データをシステム制御層経由で受け取って距離検出、姿勢検出、接触などの外的要因の検出を行う認識系処理部と、前記第１の制御部からのコマンドに基づいてロボットの機体上の動作制御を処理する出力系処理部とを含むことを特徴とする請求項１に記載のロボット制御システム。

【請求項６】前記通信部は、前記認識系処理部による検出情報を前記第１の制御部に通知するとともに、前記第１の制御部によるコマンドを前記出力系処理部に転送することを特徴とする請求項５に記載のロボット制御システム。

【請求項７】前記通信部は、前記第２の制御部からの情報を前記第１の制御部に通知する情報通信インターフェースを備えることを特徴とする請求項１に記載のロボット制御システム。

【請求項８】前記通信部は、前記第１の制御部が前記第２の制御部を制御するためのコマンド・インターフェースを備えることを特徴とする請求項１に記載のロボット制御システム。

【請求項９】前記通信部は、

前記第１の制御部が前記第２の制御部から取得したい情報を意味ベースで指定するための情報データベースを備え、

該情報データベース中の該当するレコードを登録することにより、前記第２の制御部から前記第１の制御部に対して該当する情報を転送する、ことを特徴とする請求項１に記載のロボット制御システム。

2

【請求項１０】前記通信部は、

前記第１の制御部が前記第２の制御部に対して発行したいコマンドを意味ベースで指定するためのコマンド・データベースを備え、

前記第１の制御部は前記コマンド・データベースを利用してコマンドを意味ベースで選択する、ことを特徴とする請求項１に記載のロボット制御システム。

【請求項１１】前記通信部は、前記第１の制御部による認識結果を前記第２の制御部に通知するとともに、認識結果と前記第２の制御部で可能な行動との関係を前記第１の制御部に通知するフィードバック・インターフェースを備えることを特徴とする請求項１に記載のロボット制御システム。

【請求項１２】前記第１の制御部と前記第２の制御部は独立して取り扱い可能に構成されていることを特徴とする請求項１に記載のロボット制御システム。

【請求項１３】前記通信部は前記第２の制御部が検出したシステム・イベントを前記第１の制御部に通知することを特徴とする請求項１に記載のロボット制御システム。

【請求項１４】前記通信部は、

前記第２の制御部が検出したシャットダウン要因を前記第１の制御部に通知する手段と、

前記第１の制御部が設定したシャットダウンに対するレジューム条件を前記第２の制御部に通知する手段と、

を備えることを特徴とする請求項１に記載のロボット制御システム。

【請求項１５】前記通信部は、前記第２の制御部が設定する推奨レジューム条件を前記第１の制御部に通知する手段をさらに含むことを特徴とする請求項１３に記載のロボット制御システム。

【請求項１６】複数のハードウェア構成要素の組み合わせからなるロボットの動作を、ロボットのハードウェア構成情報に依存しない処理を行う第１の制御モジュール、並びに、ロボットのハードウェア構成情報に依存する処理を行う第２の制御モジュールを用いて制御するロボット制御方法であって、

前記第１の制御モジュールと前記第２の制御モジュール間で通信を行う通信ステップを備えることを特徴とするロボット制御方法。

【請求項１７】前記第１の制御モジュールは、ロボットの構成や動作を抽象化したモデルによってロボットの行動シーケンスを決定するアプリケーション・ソフトウェアによって実装されることを特徴とする請求項１６に記載のロボット制御方法。

【請求項１８】前記第１の制御モジュールは、ロボットの感情をモデル化した感情モデルと、本能をモデル化した本能モデルと、外部事象とロボットがとる行動との因

果関係を逐次記憶していく学習モデルと、行動パターンをモデル化した行動モデルのうち少なくとも１つを備えることを特徴とする請求項１６に記載のロボット制御方法。

【請求項１９】前記第２の制御モジュールは、ロボットの機体上における基本的な機能を提供するミドルウェア・ソフトウェアによって実装されることを特徴とする請求項１６に記載のロボット制御システム。

【請求項２０】前記第２の制御モジュールは、ロボットのハードウェアから検出される入力データをシステム制御層経由で受け取って距離検出、姿勢検出、接触などの外的要因の検出を行う認識系処理モジュールと、前記第１の制御モジュールからのコマンドに基づいてロボットの機体上の動作制御を処理する出力系処理モジュールとを含むことを特徴とする請求項１６に記載のロボット制御方法。

【請求項２１】前記通信ステップでは、前記認識系処理モジュールの実行による検出情報を前記第１の制御モジュールに通知するとともに、前記第１の制御モジュールの実行によるコマンドを前記出力系処理モジュールに転送することを特徴とする請求項２０に記載のロボット制御方法。

【請求項２２】前記通信ステップでは、前記第２の制御モジュールからの情報を前記第１の制御モジュールに通知する情報通信インターフェースを利用することを特徴とする請求項１６に記載のロボット制御方法。

【請求項２３】前記通信ステップでは、前記第１の制御モジュールが前記第２の制御モジュールを制御するためのコマンド・インターフェースを利用することを特徴とする請求項１６に記載のロボット制御方法。

【請求項２４】前記通信ステップでは、
前記第１の制御モジュールが前記第２の制御モジュールから取得したい情報を意味ベースで指定するための情報データベースを利用して、
該情報データベース中の該当するレコードを登録することにより、前記第２の制御モジュールから前記第１の制御モジュールに対して該当する情報を転送する、ことを特徴とする請求項１６に記載のロボット制御方法。

【請求項２５】前記通信ステップでは、
前記第１の制御モジュールが前記第２の制御モジュールに対して発行したいコマンドを意味ベースで指定するためのコマンド・データベースを利用して、前記第１の制御モジュールはコマンドを意味ベースで選択する、ことを特徴とする請求項１６に記載のロボット制御方法。

【請求項２６】前記通信ステップは、前記第１の制御モジュールによる認識結果を前記第２の制御モジュールに通知するとともに、認識結果と前記第２の制御モジュールで可能な行動との関係を前記第１の制御モジュールに通知するフィードバック・ループを実行することを特徴とする請求項１６に記載のロボット制御方法。

【請求項２７】前記第１の制御モジュールと前記第２の制御モジュールは独立して取り扱い可能に構成されていることを特徴とする請求項１６に記載のロボット制御方法。

【請求項２８】前記通信ステップは、前記第２の制御モジュールが検出したシステム・イベントを前記第１の制御モジュールに通知するサブステップを含むことを特徴とする請求項１６に記載のロボット制御方法。

【請求項２９】前記通信ステップは、
前記第２の制御モジュールが検出したシャットダウン要因を前記第１の制御モジュールに通知するサブステップと、
前記第１の制御モジュールが設定したシャットダウンに対するレジューム条件を前記第２の制御モジュールに通知するサブステップと、を備えることを特徴とする請求項１６に記載のロボット制御方法。

【請求項３０】前記通信ステップは、前記第２の制御モジュールが設定する推奨レジューム条件を前記第１の制御モジュールに通知するサブステップをさらに含むことを特徴とする請求項２９に記載のロボット制御方法。

【請求項３１】オブジェクト指向プログラムで構成されるロボット制御システムであって、
ロボットのハードウェア構成に依存しない処理を行うアプリケーション・オブジェクトと、
ロボットのハードウェア構成に依存する処理を行うミドルウェア・オブジェクトと、
前記アプリケーション・オブジェクトからの意味ベースでのコマンドに対応する前記ミドルウェア・オブジェクトで使用される情報が登録されたインフォメーション・データベースと、
前記インフォメーション・データベースに基づいて、前記アプリケーション・オブジェクトと前記ミドルウェア・オブジェクト間での通信を制御するオブジェクト制御手段と、を具備することを特徴とするロボット制御システム。

【請求項３２】前記インフォメーション・データベースは、登録される情報の有する意味ベースの側面を階層的に記述する、ことを特徴とする請求項３１に記載のロボット制御システム。

【請求項３３】前記インフォメーション・データベースのフォーマットは、少なくともインフォメーション識別情報フィールド、分類フィールド、センサ情報識別フィールドを備える、ことを特徴とする請求項３１に記載のロボット制御システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、脚式歩行型など多関節型のロボットをソフトウェア・プログラムを用いて制御するロボット制御システム及びロボット制御方法に係り、特に、脚部や頭部など各動作ユニットの着脱・交

換などに伴ってハードウェア構成が大幅に変更する可能性がある多関節型ロボットをソフトウェア・プログラムを用いて制御するロボット制御システム及びロボット制御方法に関する。

【０００２】更に詳しくは、本発明は、ハードウェア構成に対して依存性の高いソフトウェア層とハードウェア構成に非依存のソフトウェア層の組み合わせからなるソフトウェア・プログラムを用いて多関節型ロボットを制御するロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースに係り、特に、ミドルウェアのようなハードウェア依存のソフトウェア層と、アプリケーションなどのハードウェア非依存のソフトウェア層との組み合わせを動的に変更して多関節型ロボットを制御するロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースに関する。

【０００３】

【従来の技術】電気的若しくは磁気的な作用を用いて人間の動作に似せた運動を行う機械装置のことを「ロボット」という。ロボットの語源は、スラブ語のＲＯＢＯＴＡ（奴隷機械）に由来すると言われている。わが国では、ロボットが普及し始めたのは１９６０年代末からであるが、その多くは、工場における生産作業の自動化・無人化などを目的としたマニピュレータや搬送ロボットなどの産業用ロボット（industrial robot）であった。

【０００４】最近では、イヌやネコのように４足歩行の動物の身体メカニズムやその動作を模したペット型ロボット、あるいは、ヒトやサルなどの２足直立歩行を行う動物の身体メカニズムや動作を模した「人間形」若しくは「人間型」のロボット（humanoid robot）など、脚式移動ロボットの構造やその安定歩行制御に関する研究開発が進展し、実用化への期待も高まってきている。これら脚式移動ロボットは、クローラ式ロボットに比し不安定で姿勢制御や歩行制御が難しくなるが、階段の昇降や障害物の乗り越え等、柔軟な歩行・走行動作を実現できるという点で優れている。

【０００５】アーム式ロボットのように、ある特定の場所に植設して用いるような据置きタイプのロボットは、部品の組立・選別作業など固定的・局所的な作業空間でのみ活動する。これに対し、移動式のロボットは、作業空間は非限定的であり、所定の経路上または無経路上を自在に移動して、所定の若しくは任意の人的作業を代行したり、ヒトやイヌあるいはその他の生命体に置き換わる種々のサービスを提供することができる。

【０００６】脚式移動ロボットの用途の１つとして、産業活動・生産活動等における各種の難作業の代行が挙げられる。例えば、原子力発電プラントや火力発電プラント、石油化学プラントにおけるメンテナンス作業、製造工場における部品の搬送・組立作業、高層ビルにおける清掃、火災現場その他における救助といったような危険作業・難作業の代行などである。

【０００７】また、脚式移動ロボットの他の用途として、上述の作業支援というよりも、生活密着型、すなわち人間との「共生」あるいは「エンターティンメント」という用途が挙げられる。この種のロボットは、ヒトあるいはイヌ（ペット）などの比較的知性の高い脚式歩行動物の動作メカニズムや四肢を利用した豊かな感情表現をエミュレートする。また、あらかじめ入力された動作パターンを単に忠実に実行するだけではなく、ユーザ（あるいは他のロボット）から受ける言葉や態度（「褒める」とか「叱る」、「叩く」など）に対して動的に対応した、生き生きとした応答表現を実現することも要求される。

【０００８】従来の玩具機械は、ユーザ操作と応答動作との関係が固定的であり、玩具の動作をユーザの好みに合わせて変更することはできない。この結果、ユーザは同じ動作しか繰り返さない玩具をやがては飽きてしまうことになる。

【０００９】これに対し、知能型のロボットは、動作に起因する行動モデルや学習モデルを備えており、外部からの音声や画像、触覚などの入力情報に基づいてモデルを変化させて動作を決定することにより、自律的な思考及び動作制御を実現する。ロボットが感情モデルや本能モデルを用意することにより、ロボット自身の感情や本能に従った自律的な行動を表出することができる。また、ロボットが画像入力装置や音声入出力装置を装備し、画像認識処理や音声認識処理を行うことにより、より高度な知的レベルで人間とのリアリスティックなコミュニケーションを実現することも可能となる。

【００１０】昨今の脚式移動ロボットは高い情報処理能力を備えており、インテリジェントなロボットそのものを一種の計算機システムとして捉えることができる。

【００１１】例えば、ロボットは、感情モデルや行動モデル、学習モデルなどのように、動作に関する各種の規則をモデル化して保持しており、これら各モデルに従ってユーザ・アクションなどの外的要因に応答した行動計画を立案して、各関節アクチュエータの駆動や音声出力などを介して行動計画を体現し、ユーザ・フィードバックすることができる。このような行動計画の立案やこれを機体上で体現するためのロボットの動作制御は、計算機システム上におけるプログラム・コード（例えばアプリケーションなど）の実行という形態で実装される。

【００１２】一般的な計算機システムとロボットとの主な相違として、前者はシステムを構成するハードウェア・コンポーネントの種類や組み合わせ（すなわちハードウェア構成）が各システム間で差が少ないのに対して、後者はハードウェア構成がシステム間で大幅に変更するという点を挙げることができよう。例えば、ひとえに移動ロボットといっても、胴体に対して取り付けられる可動部として、頭と脚部と尻尾で構成されるロボットや、

胴体と車輪のみで構成されるロボットなど、千差万別である。

【００１３】装備されるハードウェア構成がシステム間で比較的均一である計算機システムにおいては、システム上で実行するソフトウェアのデザインはハードウェアの影響を比較的受けずに済む。これに対して、後者のロボットの場合には、特にハードウェア操作を行うミドルウェアのような制御ソフトウェア層においてはハードウェア依存性が極めて高くなる。

【００１４】例えば、ロボットの移動制御を考えた場合、移動手段が可動脚の場合と車輪の場合と、２足と４足の場合とでは、移動時・歩行時における安定度判別規範がまったく相違するので、アプリケーションを実行するための動作環境はシステム間で大幅に異なる。

【００１５】ロボットのソフトウェア開発を考えた場合、このような事情を鑑み、ハードウェアの依存性が比較的低いソフトウェア層と、ハードウェアの依存性が高いソフトウェア層とに区別することが効率的と思料される。すなわち、ハードウェア非依存ソフトウェアと、ハードウェア依存ソフトウェアとを個別に開発しておき、両者の組み合わせを変えることにより、多岐にわたる、さまざまな特性や性能を持つ製品ラインアップを提供することができる。

【００１６】ハードウェア非依存のソフトウェアは、例えば、感情モデルや行動モデル、学習モデルなどのハードウェア操作との関係が少ない処理を行うアプリケーション層ソフトウェアである。また、ハードウェア依存のソフトウェアは、例えば、ロボット１の基本的な機能を提供するソフトウェア・モジュールの集まりで構成されるミドルウェア層ソフトウェアであり、各モジュールの構成はロボットの機械的・電気的な特性や仕様、形状などハードウェア属性の影響を受ける。ミドルウェアは、機能的には、各部のセンサの入力を処理・認識して上位のアプリケーションに通知する認識系のミドルウェアと、アプリケーションが発行するコマンドに従って各関節アクチュエータの駆動などハードウェアの駆動制御を行う出力系のミドルウェアに大別することができる。

【００１７】例えば、ハードウェア構成に適合したミドルウェアをロボットに導入することによって、同じアプリケーションを様々なハードウェア構成のロボット上で実行可能となる。

【００１８】ところで、ロボットを始めとして各種の計算機システムに対してソフトウェアを導入する形態として、新しいソフトウェアをリムーバブル・メディアを介して供給したり、あるいはネットワーク経由でソフトウェアをダウンロードすることが挙げられる。例えば、ロボットの本体のある部位に、メモリ・カードやメモリ・スティックなどのリムーバブル・メディアを装填するためのメモリ・スロットを配設しておくことにより、リムーバブル・メディアをスロットに対して挿脱する作業だけで、アプリケーションやミドルウェアなどの新しいソフトウェアを簡単にロボットに導入することができる。

【００１９】複数のソフトウェア層で構成されるロボットの制御システムに新規ソフトウェアを導入するに際しては、新たに導入されるソフトウェアが他のソフトウェア層との相性、すなわち互換性が保たれいてる必要がある。

【００２０】より具体的に言うならば、アプリケーションとミドルウェア間の任意の組み合わせを許容するためには、これらソフトウェア層間でのデータやコマンドを交換する形式、すなわちプログラム間のインターフェースを確立させておく必要がある。

【００２１】

【発明が解決しようとする課題】本発明の目的は、脚式歩行型など多関節型のロボットをソフトウェア・プログラムを用いて制御する、優れたロボット制御システム及びロボット制御方法を提供することにある。

【００２２】本発明の更なる目的は、脚部や頭部など各動作ユニットの着脱・交換などに伴ってハードウェア構成が大幅に変更する可能性がある多関節型ロボットをソフトウェア・プログラムを用いて制御する、優れたロボット制御システム及びロボット制御方法を提供することにある。

【００２３】本発明の更なる目的は、ハードウェア構成に対して依存性の高いソフトウェア層とハードウェア構成に非依存のソフトウェア層の組み合わせからなるソフトウェア・プログラムを用いて多関節型ロボットを制御する、優れたロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースを提供することにある。

【００２４】本発明の更なる目的は、ミドルウェアのようなハードウェア依存のソフトウェア層と、アプリケーションなどのハードウェア非依存のソフトウェア層との組み合わせを動的に変更して多関節型ロボットを制御する、優れたロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースを提供することにある。

【００２５】

【課題を解決するための手段及び作用】本発明は、上記課題を参酌してなされたものであり、その第１の側面は、複数のハードウェア構成要素の組み合わせからなるロボットの動作を制御するロボット制御システムであって、ロボットのハードウェア構成情報に依存しない処理を行う第１の制御部とロボットのハードウェア構成情報に依存する処理を行う第２の制御部と前記第１及び第２の制御部間の通信を行う通信部と、を具備することを特徴とするロボット制御システムである。

【００２６】但し、ここで言う「システム」とは、複数の装置（又は特定の機能を実現する機能モジュール）が論理的に集合した物のことを言い、各装置や機能モジュ

ールが単一の筐体内にあるか否かは特に問わない。

【００２７】ここで言う第１の制御部は、ハードウェア構成に非依存のアプリケーション層ソフトウェアによって実装される。また、第２の制御部は、ハードウェア構成に対する依存性の高いミドルウェア層ソフトウェアによって実装される。また、通信部は、アプリケーションとミドルウェアの間のデータ交換処理を実現するプログラム・インターフェースという形態で実装することができる。

【００２８】本発明の第１の側面に係るロボット制御システムによれば、通信部が、アプリケーション層とミドルウェア層の間のインターフェースとなり、ソフトウェア層間でのデータやコマンドを交換する形式を確立することによって、アプリケーションとミドルウェア間の任意の組み合わせを許容することができる。

【００２９】前記第１の制御部は、例えば、ロボットの構成や動作を抽象化したモデルによってロボットの行動シーケンスを決定するアプリケーション・ソフトウェアを実行することにより実現される。アプリケーション・ソフトウェアは、例えば、ロボットの感情をモデル化した感情モデル、本能をモデル化した本能モデル、外部事象とロボットがとる行動との因果関係を逐次記憶していく学習モデル、行動パターンをモデル化した行動モデルなどを備えている。

【００３０】また、前記第２の制御部は、例えば、ロボットの機体上における基本的な機能を提供するミドルウェア・ソフトウェアを実行することにより実現される。ミドルウェア・ソフトウェアは、例えば、ロボットのハードウェアから検出される入力データをシステム制御層経由で受け取って距離検出、姿勢検出、接触などの外的要因の検出をハードウェアの構成を考慮して行う認識系処理部と、アプリケーションからのコマンドに基づいてロボットの機体上の動作制御を処理する出力系処理部とで構成される。

【００３１】前記通信部は、前記認識系処理部による検出情報を前記第１の制御部に通知するとともに、前記第１の制御部によるコマンドを前記出力系処理部に転送する。

【００３２】また、前記通信部は、前記第２の制御部からの情報を前記第１の制御部に通知する情報通信インターフェースや、前記第１の制御部が前記第２の制御部を制御するためのコマンド・インターフェースなどを備えている。

【００３３】また、前記通信部は、前記第１の制御部が前記第２の制御部から取得したい情報を意味ベースで指定するための情報データベースを備えていてもよい。このような場合、該情報データベース中の該当するレコードを登録することにより、前記第２の制御部から前記第１の制御部に対して該当する情報を転送することができる。

【００３４】また、前記通信部は、前記第１の制御部が前記第２の制御部に対して発行したいコマンドを意味ベースで指定するためのコマンド・データベースを備えていてもよい。このような場合、前記第１の制御部は前記コマンド・データベースを利用してコマンドを意味ベースで選択することができる。

【００３５】また、前記通信部は、前記第１の制御部による認識結果を前記第２の制御部に通知するとともに、認識結果と前記第２の制御部で可能な行動との関係を前記第１の制御部に通知するためのフィードバック・インターフェースを備えていてもよい。

【００３６】また、前記第１の制御部と前記第２の制御部は独立して取り扱い可能に構成されていてもよい。

【００３７】また、前記通信部は前記第２の制御部が検出したシステム・イベントを前記第１の制御部に通知するようにしてもよい。

【００３８】また、前記通信部は、前記第２の制御部が検出したシャットダウン要因を前記第１の制御部に通知する手段と、前記第１の制御部が設定したシャットダウンに対するレジューム条件を前記第２の制御部に通知する手段とを備えていてもよい。また、前記第２の制御部が設定する推奨レジューム条件を前記第１の制御部に通知する手段をさらに含んでいてもよい。

【００３９】本発明の第１の側面に係るロボット制御システムによれば、ロボットのハードウェア構成に依存するミドルウェア層と、ハードウェア構成に依存しないアプリケーション層の間に、意味的に動作を行うためのインターフェースとデータベースを用意することによって、ロボット上に導入するミドルウェアとアプリケーションの組み合わせを変更しても、常に正常な動作を保証することができる。また、アプリケーションは、ミドルウェアを介して適当な入力データを取得したり、適切なコマンドを発行することができる。

【００４０】また、本発明の第２の側面は、複数のハードウェア構成要素の組み合わせからなるロボットの動作を、ロボットのハードウェア構成情報に依存しない処理を行う第１の制御モジュール、並びに、ロボットのハードウェア構成情報に依存する処理を行う第２の制御モジュールを用いて制御するロボット制御方法であって、前記第１の制御モジュールと前記第２の制御モジュール間で通信を行う通信ステップを備えることを特徴とするロボット制御方法である。

【００４１】ここで言う第１の制御モジュールは、ハードウェア構成に非依存のアプリケーション層ソフトウェアによって実装される。また、第２の制御モジュールは、ハードウェア構成に対する依存性の高いミドルウェア層ソフトウェアによって実装される。また、通信ステップは、アプリケーションとミドルウェアの間のデータ交換処理を実現するプログラム・インターフェースという形態で実装することができる。

【００４２】本発明の第２の側面に係るロボット制御方法によれば、通信ステップにおいて、アプリケーション層とミドルウェア層の間のインターフェースを実現して、ソフトウェア層間でのデータやコマンドを交換する形式を確立することによって、アプリケーションとミドルウェア間の任意の組み合わせを許容することができる。

【００４３】前記第１の制御モジュールは、前記第１の制御モジュールは、ロボットの構成や動作を抽象化したモデルによってロボットの行動シーケンスを決定するアプリケーション・ソフトウェアによって実装される。アプリケーション・ソフトウェアは、例えば、ロボットの感情をモデル化した感情モデル、本能をモデル化した本能モデル、外部事象とロボットがとる行動との因果関係を逐次記憶していく学習モデル、行動パターンをモデル化した行動モデルなどで構成される。

【００４４】また、前記第２の制御モジュールは、ロボットの機体上における基本的な機能を提供するミドルウェア・ソフトウェアによって実装される。ミドルウェアは、例えば、ロボットのハードウェアから検出される入力データをシステム制御層経由で受け取って距離検出、姿勢検出、接触などの外的要因の検出を行う認識系処理モジュールと、アプリケーションからのコマンドに基づいてロボットの機体上の動作制御を処理する出力系処理モジュールとで構成される。

【００４５】前記通信ステップでは、前記認識系処理モジュールの実行による検出情報を前記第１の制御モジュールに通知するとともに、前記第１の制御モジュールの実行によるコマンドを前記出力系処理モジュールに転送するようにしてもよい。

【００４６】また、前記通信ステップでは、前記第２の制御モジュールからの情報を前記第１の制御モジュールに通知する情報通信インターフェースを利用するようにしてもよい。

【００４７】また、前記通信ステップでは、前記第１の制御モジュールが前記第２の制御モジュールを制御するためのコマンド・インターフェースを利用するようにしてもよい。

【００４８】また、前記通信ステップでは、前記第１の制御モジュールが前記第２の制御モジュールから取得したい情報を意味ベースで指定するための情報データベースを利用するようにしてもよい。このような場合、該情報データベース中の該当するレコードを登録することにより、前記第２の制御モジュールから前記第１の制御モジュールに対して該当する情報を転送することができる。

【００４９】また、前記通信ステップでは、前記第１の制御モジュールが前記第２の制御モジュールに対して発行したいコマンドを意味ベースで指定するためのコマンド・データベースを利用して、前記第１の制御モジュールはコマンドを意味ベースで選択するようにしてもよい。

【００５０】また、前記通信ステップは、前記第１の制御モジュールによる認識結果を前記第２の制御モジュールに通知するとともに、認識結果と前記第２の制御モジュールで可能な行動との関係を前記第１の制御モジュールに通知するフィードバック・ループを実行するようにしてもよい。

【００５１】また、前記第１の制御モジュールと前記第２の制御モジュールは独立して取り扱い可能に構成するようにしてもよい。

【００５２】また、前記通信ステップは、前記第２の制御モジュールが検出したシステム・イベントを前記第１の制御モジュールに通知するサブステップを含んでもよい。

【００５３】また、前記通信ステップは、前記第２の制御モジュールが検出したシャットダウン要因を前記第１の制御モジュールに通知するサブステップと、前記第１の制御モジュールが設定したシャットダウンに対するレジューム条件を前記第２の制御モジュールに通知するサブステップとを備えていてもよい。また、前記通信ステップは、前記第２の制御モジュールが設定する推奨レジューム条件を前記第１の制御モジュールに通知するサブステップをさらに含んでいてもよい。

【００５４】本発明の第２の側面に係るロボット制御方法によれば、ロボットのハードウェア構成に依存するミドルウェア層と、ハードウェア構成に依存しないアプリケーション層の間に、意味的に動作を行うためのインターフェースとデータベースを用意することによって、ロボット上に導入するミドルウェアとアプリケーションの組み合わせを変更しても、常に正常な動作を保証することができる。また、アプリケーションは、ミドルウェアを介して適当な入力データを取得したり、適切なコマンドを発行することができる。

【００５５】また、本発明の第３の側面は、オブジェクト指向プログラムで構成されるロボット制御システムであって、ロボットのハードウェア構成に依存しない処理を行うアプリケーション・オブジェクトと、ロボットのハードウェア構成に依存する処理を行うミドルウェア・オブジェクトと、前記アプリケーション・オブジェクトからの意味ベースでのコマンドに対応する前記ミドルウェア・オブジェクトで使用される情報が登録されたインフォメーション・データベースと、前記インフォメーション・データベースに基づいて、前記アプリケーション・オブジェクトと前記ミドルウェア・オブジェクト間での通信を制御するオブジェクト制御手段と、を具備することを特徴とするロボット制御システムである。

【００５６】本発明の第３の側面に係るロボット制御システムによれば、オブジェクト制御手段は、インフォメーション・データベースを参照することによってアプリ

ケーション層とミドルウェア層の間のインターフェースを形成して、ソフトウェア層間でのデータやコマンドを交換する形式を確立することによって、アプリケーションとミドルウェア間の任意の組み合わせを実現することができる。

【００５７】ここで、前記インフォメーション・データベースは、登録される情報の有する意味ベースの側面を階層的に記述している。

【００５８】前記インフォメーション・データベースのフォーマットは、少なくともインフォメーション識別情報フィールド、分類フィールド、センサ情報識別フィールドを備えている。

【００５９】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００６０】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【００６１】図１には、本発明の実施に供されるロボットのハードウェア構成を模式的に図解している。同図に示すように、ロボットのハードウェアは、制御系サブシステム１０と、駆動系サブシステム５０とで構成される。

【００６２】ロボットの制御系サブシステムは、ＣＰＵ（Central Processing Unit）１１と、メイン・メモリ１２と、固定型メモリ装置１３と、交換可能メモリ装置１４とで構成され、る。

【００６３】メイン・コントローラとしてのＣＰＵ１１は、システム制御ソフトウェアの制御下で、アプリケーションのようなハードウェア非依存プログラムや、ミドルウェアのようなハードウェア依存プログラムを実行して、ロボットという装置全体の動作を統括的に制御するようになっている。

【００６４】ＣＰＵ１１は、メモリやその他の各回路コンポーネントや周辺機器とバス接続されている。バス上の各装置にはそれぞれに固有のアドレス（メモリ・アドレス又はＩ／Ｏアドレス）が割り当てられており、ＣＰＵ１１はアドレス指定することでバス上の特定の装置と通信することができる。バスは、アドレス・バス、データ・バス、コントロール・バスを含んだ共通信号伝送路である。

【００６５】メイン・メモリ１２は、通常、複数個のＤＲＡＭ（Dynamic Random Access Memory）チップで構成される揮発性記憶装置で構成され、ＣＰＵ１１の実行プログラム・コードをロードしたり、その作業データの一時的な保存のために利用される。本実施例では、固定型メモリ装置１３や交換可能メモリ装置１４から供給されるアプリケーションやミドルウェアなどのプログラム・コードは、メイン・メモリ１２上に展開すなわちメモリ空間上にマッピングされる。

【００６６】固定型メモリ装置１３は、ロボット本体に対して固定的に取り付けられた、交換不能な不揮発性記憶装置である。例えば、フラッシュ・メモリのように、書き込み電圧の印加によりプログラマブルな不揮発性メモリ素子を用いて固定型メモリ装置１３に構成することができる。

【００６７】固定型メモリ装置１３は、ロボットの動作や思考を制御するためのアプリケーションや、ハードウェア操作用のミドルウェアなどのプログラム・コードを格納するために利用される。但し、固定型メモリ装置１３は、装置に対して固定的に設置されることから、ミドルウェアなどのハードウェア依存型のソフトウェアに関しては、ロボットの出荷時（デフォルト時）又は標準的なハードウェア構成に適合するバージョンを固定型メモリ装置１３内に用意しておくことが好ましい。

【００６８】交換可能メモリ装置１４は、ロボットに対して着脱・交換可能に取り付けられる、不揮発性記憶装置である。例えば、メモリ・カードやメモリ・スティックのようなカートリッジ式の記憶媒体を用いて交換可能メモリ装置１４を構成して、所定のメモリ・スロット上に装填することによって、機体上で交換可能な使用に供される。

【００６９】交換可能メモリ装置１４は、固定型メモリ装置１３と同様に、ロボットの動作や思考を制御するためのアプリケーションや、ハードウェア操作用のミドルウェアなどのプログラム・コードを格納するために利用される。但し、交換可能メモリ装置１４は、ロボット本体に対して着脱・交換に提供され、またハードウェア構成の異なる機種間を移動して使用に供されることが想定されるので、最新のソフトウェアを機体に提供する際などに利用することができる。

【００７０】ミドルウェアなどのハードウェア依存型のソフトウェアに関しては、ロボットの出荷時（デフォルト時）又は標準的なハードウェア構成に適合するバージョンか否かを特に意識して交換可能メモリ装置１４上に格納する必要性が低い。むしろ、アプリケーションが想定するハードウェア構成に対して動作環境を提供することができるミドルウェアを、アプリケーションと組にして、交換可能メモリ装置１４に格納することが好ましい。

【００７１】一方、ロボットの駆動系サブシステム５０は、各関節アクチュエータやその駆動制御回路、動作検出用のエンコーダ、さらにはカメラや接触センサなどの各種センサ類（いずれも図示しない）などで構成される。図示の例では、駆動系サブシステム５０は、頭部、胴体部、脚部など各駆動ユニット単位で扱われるものとする。

【００７２】さらに、駆動ユニットの少なくとも一部は、機体に対する着脱・交換などにより動的に再構成可能な物理コンポーネント（ＣＰＣ：Configurable Physi

cal Component）として構成されているものとする。

【００７３】本実施例では、各物理コンポーネントは、固有の識別情報すなわちコンポーネントＩＤが付与されている。制御系サブシステム１０のＣＰＵ１１（より具体的にはＣＰＵ１１上で実行されるシステム制御ソフトウェア）は、装備されている各物理コンポーネントに対してバス経由でアクセスして、各物理コンポーネントに対して制御コマンドを転送したり、それぞれのコンポーネントＩＤを取得することができるようになっている。検出されたコンポーネントＩＤの組み合わせが、ロボットにおける現在のハードウェア構成情報となる。

【００７４】図２には、ＣＰＣコンポーネントの着脱・交換により駆動系サブシステム５０の構成を変更した例を示している。同図（ａ）では、胴体に対して頭と脚と尻尾などの複数の物理コンポーネントが装着されているが（すなわち、ハードウェア構成情報は｛胴体ＩＤ，頭ＩＤ，脚ＩＤ，尻尾ＩＤ｝となる）、同図（ｂ）では、胴体に対して物理コンポーネントとして車輪しか装備されていない（すなわち、ハードウェア構成情報は｛胴体ＩＤ，車輪ＩＤ｝となる）。

【００７５】図２（ａ）及び図２（ｂ）に示す例のようにハードウェア構成が著しく相違するロボットの間では、同じハードウェア依存ソフトウェアを使用することができない。例えば、頭や尻尾からのセンサ入力を行うミドルウェアは、図２（ｂ）に示すハードウェア構成の装置上では動作することができない。同様に、移動手段として脚部を駆動制御するようにデザインされたミドルウェアを、図２（ｂ）に示すハードウェア構成の装置上で使用することはできない。

【００７６】次いで、ロボット制御用のソフトウェアの構成について、図３を参照しながら説明する。

【００７７】ロボットの制御用ソフトウェアは、ロボットのハードウェア構成に依存しないアプリケーション層と、ハードウェア構成に依存するミドルウェア層と、最下層のデバイス・ドライバで構成される。各層のソフトウェアは、所定のオペレーティング・システム（ＯＳ）の制御下で、ロボットのハードウェアすなわち後述する駆動系サブシステム５０の制御を行う。

【００７８】本実施例では、各層のソフトウェアのデザインには、オブジェクト指向プログラミングを採り入れることができる。オブジェクト指向における各ソフトウェアは、データとそのデータに対する処理手続きとを一体化させた「オブジェクト」というモジュール単位で扱われる。

【００７９】アプリケーション層とミドルウェア層とは、所定のプログラミング・インターフェース（以下では、「アプリケーション・インターフェース」と呼ぶことにする）を介してデータ通信が行われる。本実施例では、交換可能メモリ装置１４を介したソフトウェアの導入などによりアプリケーションとミドルウェアのさまざまな組み合わせが許容される。それぞれのアプリケーションとミドルウェアとの間の互換性は、このアプリケーション・インターフェースによって実現される。但し、アプリケーション・インターフェースの仕組みについては、後に詳解する。

【００８０】アプリケーション層とミドルウェア層は、それぞれ複数のオブジェクト・ファイルで構成される。本実施例では、アプリケーション層とミドルウェア層は、オブジェクト管理システムの制御下に置かれている。また、ミドルウェア層と、最下層のデバイス・ドライバとの間のデータ通信は、所定のプログラミング・インターフェース（以下では、「デバイス・ドライバ・インターフェース」と呼ぶことにする）を介してデータ通信が行われる。

【００８１】ハードウェアの操作、すなわち、駆動用の各関節アクチュエータへの制御指令の発行や、各センサにおける検出値の入力などは、ミドルウェアが直接行わず、各ハードウェア・コンポーネントに対応するそれぞれのデバイス・ドライバを介して行われる。

【００８２】アプリケーションは、ロボットの感情をモデル化した感情モデルと、本能をモデル化した本能モデルと、外部事象とロボットがとる行動との因果関係を逐次記憶していく学習モデルと、行動パターンをモデル化した行動モデルとを備えており、センサ入力情報すなわち外部要因を基に行動モデルによって決定された行動の出力先を切り替えるようになっている。

【００８３】感情モデルと本能モデルは、それぞれ認識結果と行動履歴を入力に持ち、感情値と本能値を管理している。行動モデルは、これら感情値や本能値を参照することができる。また、学習モデルは、外部（オペレータ）からの学習教示に基づいて行動選択確率を更新して、更新内容を行動モデルに供給する。

【００８４】アプリケーションは、ロボットの構成や動作を抽象化したモデルによって演算処理を行うので、ハードウェア属性の影響を受けないハードウェア非依存型のソフトウェアである。

【００８５】ミドルウェア層は、ロボットの機体上における基本的な機能を提供するソフトウェア・モジュールの集まりであり、各モジュールの構成はロボットの機械的・電気的な特性や仕様、形状などハードウェア属性の影響を受けるハードウェア依存型のソフトウェアである。

【００８６】ミドルウェア層は、機能的に、認識系のミドルウェアと、出力系のミドルウェアに分けることができる。

【００８７】認識系のミドルウェアでは、画像データや音声データ、その他のセンサから得られる検出データなど、ハードウェアからの生データをシステム制御層経由で受け取ってこれらを処理する。すなわち、各種入力情報に基づき、音声認識、距離検出、姿勢検出、接触、動

き検出、色認識などの処理を行い、認識結果を得る（例えば、ボールを検出した、転倒を検出した、撫でられた、叩かれた、ドミソの音階が聞こえた、動く物体を検出した、障害物を検出した、障害物を認識した、など）。認識結果は、上位のアプリケーション層に通知され、行動計画の立案などに利用される。

【００８８】一方、出力系のミドルウェアでは、歩行、動きの再生、出力音の合成、目に相当するＬＥＤの点灯制御などの機能を提供する。すなわち、アプリケーション層において立案された行動計画に関連するコマンドを受け取って、ロボットの各機能毎にロボットの各ジョイントのサーボ指令値や出力音、出力光（ＬＥＤ）、出力音声などを生成して、出力すなわち仮想ロボットを介してロボット上で実演する。このような仕組みにより、アプリケーション側からは抽象的な行動コマンド（例えば、前進、後退、喜ぶ、吼える、寝る、体操する、驚く、トラッキングするなど）を与えることで、ロボットの各関節アクチュエータや機体上のその他の出力部による動作を制御することができる。

【００８９】本実施例に係るミドルウェアは、アプリケーション～ミドルウェア間の意味ベースでの情報通信を用意し、知識ベースを共有することができる。データベースには、アプリケーションが意味ベースで入力情報を特定するために使用する情報データベース（InformationDB）と、アプリケーションが意味ベースで実行コマンドを選択するために使用するコマンド・データベース（CommandDB）が含まれる。データベースの詳細については後述に譲る。

【００９０】これらロボット制御用の各層ソフトウェアは、固定型メモリ装置１３や交換可能メモリ装置１４によってロボット１の機体内に提供される。各層のソフトウェアは、メイン・メモリ１２上に展開されて、すなわちメモリ空間上にマッピングして使用される。

【００９１】本実施例では、ロボットのハードウェア構成に依存するミドルウェア層と、行動選択などハードウェア構成に依存しないアプリケーション層間にアプリケーション・インターフェースを介在させて規定する。これにより、以下の機能を満足することができる。

【００９２】（１）ミドルウェアとアプリケーションを独立して取り扱い可能に構成することで、１つのアプリケーションで様々なハードウェア構成のロボットに対応できるようにする。（ユーザは、自分が所有する（あるいは育てた）アプリケーションを機体間で移動させたり、流通・販売することができる。）

【００９３】（２）ミドルウェアとアプリケーションを独立して取り扱い可能に構成することで、ある１つのハードウェア構成を持つ機体上で、さまざまなアプリケーションに対応するミドルウェアを作成することができる。ミドルウェアの再利用性が高くなるので、アプリケーションの開発効率が向上する。

【００９４】（３）ミドルウェアとアプリケーションを独立して取り扱い可能に構成することで、同一のハードウェア構成を持つ機体上で、同一のアプリケーションであっても、ミドルウェアを変更することで、実際の表現力や制御性の向上を実現することができる。

【００９５】（４）バイナリ互換を保証することで、アプリケーション又はミドルウェアのバイナリ・コードをメモリ・スティックなどの交換可能メモリ装置１４などから機体にダウンロードすることによって、簡単にソフトウェアを入れ替えて、上述した利点を享受することができる。しかも、コンパイル作業なしでソフトウェアを交換することができる。

【００９６】（５）アプリケーションとミドルウェアが独立しているので、ソフトウェア開発ベンダは、得意な領域の機能開発に集中することができる。例えば、制御系を専門とするベンダはミドルウェアの開発に注力することにより、色々なアプリケーションを利用することができる。

【００９７】本実施例に係るアプリケーション・インターフェースは、以下の機能実現手段を備えている。

【００９８】（１）ミドルウェアで検出した情報をアプリケーションに通知する情報通信インターフェース（SensorInformation＋SensorValueVector/SystemInfo/Boot/Shootdown）。

【００９９】（２）アプリケーションがミドルウェアを制御するためのコマンド・インターフェース（Command/Status）。

【０１００】（３）アプリケーションで検出した認識結果をミドルウェアに通知して、認識結果とミドルウェアで可能な行動との関係をアプリケーションに通知するためのフィードバック・インターフェース（TargetInfo/CommandInfo）。

【０１０１】（４）情報通信インターフェースで供給されるデータの中で、アプリケーションで使用する情報を意味ベースで選択するための手段。この意味ベース選択手段によれば、アプリケーションは、情報供給源たるセンサの実際の場所を意識することなく、学習のための入力情報（例えば、ユーザから「なでられた」など）を取得することができる。また、期待していたセンサやスイッチなどのハードウェアが機体上に実在しない場合であっても、代替のセンサを選択して（例えば、「なでられた」ことを知覚するために、頭に配設されている何らかのスイッチを代用するなど）、該当する感情（例えば、快の感情）に結びつけることなどが可能となる。ミドルウェアは、意味ベースでの入力情報の特定のために、情報データベースInformationDB（後述）を用意する。アプリケーションは、InformationDBに使用する情報を登録することにより意味ベースでの入力情報の取得が実現される。

【０１０２】（５）上記のコマンド・インターフェース

でサポートされるコマンドの中で、アプリケーションで実行したいコマンドを意味ベースで選択する手段。この意味ベース選択手段によれば、アプリケーションは、ミドルウェアが実際にどのようなモータ／エフェクタを作動させるのかを意識することなく、必要な行動を実行させることができる。ミドルウェアは、意味ベースでのコマンド選択を実現するために、コマンド・データベースCommandDB（後述）を用意する。

【０１０３】（６）アプリケーションで認識を行うのに必要な、実行時に動的に変化するロボットの情報を提供する手段。例えば、ロボットの姿勢や移動速度に関する情報を提供するために、SensorInformation中にHeaderInfo領域（後述）を規定する。

【０１０４】（７）アプリケーションで認識を行うのに必要な、実行時に動的に変化するセンサの物理情報を提供する手段。例えば、センサがどちらの方向を向いているかを通知するために、SensorInformation中のHeaderInfo領域並びにSensorInfo領域を使用する。

【０１０５】（８）アプリケーションで移動の行動を実現するのに必要な、ロボットの大きさ・移動速度などの物理情報を提供する手段。例えば、検出した隙間を現在のハードウェア構成を持つロボットが通過することができるかなど、CommandのTargetPos・アプリケーション層への情報提供を行う。

【０１０６】（９）構成によらない反射ループを構成するために、情報通信データから入力された情報に連動するあらかじめ準備された行動をコマンド・インターフェースで通知する手段。Interaction系の分類を準備して、入力情報のSensorIDをKeyとしてCommandDBを検索する。

【０１０７】（10）ミドルウェアの都合によるコマンドの実行をアプリケーションに依頼する手段。例えば、転倒からの復帰動作などのような、ミドルウェアの構造に依存して変化する処理を、アプリケーション層が詳細を理解する必要なく処理することを可能にすることで、ミドルウェアとアプリケーションの状態の整合性をとることができる。転倒に関して付言すれば、どういう姿勢を経由すると安全に立つことができるかなどの転倒姿勢からの復帰方法は、ミドルウェアがサポートする姿勢の種類によって相違する。勿論、ミドルウェアの性能にも依存する。

【０１０８】（11）シャットダウン（Shutdown）とレジューム（Resume）条件を通知する手段。シャットダウンの要因とレジュームの条件は密接に影響する。例えば、放電時のバッテリの温度が上昇し過ぎた場合に、温度が適温まで降下する時間はハードウェアの放熱性能など機体の特性に依存する。ミドルウェアは、その際のレジューム条件として、次に起きてもよい時間を推奨レジューム条件としてアプリケーションに通知する。アプリケーションは、推奨条件とアプリケーション自身の都合から、最終的なレジューム条件を設定して、シャットダウン時にミドルウェアに通知する。（Shutdown/Resume Condition）

【０１０９】（12）アプリケーションでの認識結果をミドルウェアの制御で利用する手段。アプリケーション層で認識結果を出力サービスをTrackingコマンドに追加することで、ミドルウエアのTracking担当ソフトウェア・モジュールは、認識結果サービスの中から指定された認識結果をアプリケーション層でコマンド決定しているソフトウェアを介さずに獲得することができるので、処理が高速化する。（TargetInfo/CommandのTargetID）

【０１１０】（13）ハードウェア構成によって変更するパラメータをシステムが自動検索してミドルウェアに通知する手段。例えば、新規のハードウェア構成となっても、パラメータ・ファイルを入れ替えるだけでソフトウェアの変更が不要となる。物理コンポーネント上に実装されたＲＯＭをシステムが検索して情報を通知する場合もある。この手段が提供する機能により、汎用性が向上し、バイナリ互換にも影響する。

【０１１１】以下では、アプリケーション・インターフェースの代表的な機能実現手段について詳解する。

【０１１２】インフォメーション・データベース（InformationDB）

本実施例では、アプリケーション・インターフェース経由で供給されるデータの中で、アプリケーションで使用する情報を意味ベースで選択することができる。そのために、ミドルウェアは、インフォメーション・データベースを用意する。アプリケーションは、使用する情報をインフォメーション・データベースに意味ベースで登録することにより、必要な入力情報の取得を行うことが可能となる。

【０１１３】アプリケーションは、情報供給源たるセンサの実際の場所を意識することなく、学習のための入力情報（例えば、ユーザから「なでられた」など）を取得することができる。また、期待していたセンサやスイッチなどのハードウェアが機体上に実在しない場合であっても、代替のセンサを選択して（例えば、「なでられた」ことを知覚するために、頭に配設されている何らかのスイッチを代用するなど）、該当する感情（例えば、快の感情）に結びつけることなどが可能となる。

【０１１４】図４には、インフォメーション・データベースのフォーマット例を示している。同図に示すように、インフォメーション・データベースの１つのレコードは、インフォメーション識別情報（InformationID）フィールドと、分類フィールドと、センサ値情報フィールドと、センサ識別情報（SensorID）フィールドで構成される。

【０１１５】インフォメーション識別情報（InformationID）は、レコードに一義に割り振られる固有の識別子であり、ミドルウェアが勝手に決めることができる。

【０１１６】分類フィールドは、さらに、互換、大、中、小の各フィールドに細分化される。互換フィールドは、その後に書かれるＩＤの意味が標準又は拡張のＩＤのいずれであるかを示すフラグである。大、中、小の各フィールドには大分類、中分類、小分類がそれぞれ記入される。

【０１１７】分類は、意味ベースで情報を選択することを意識して、すなわち情報の持つ意味ベースの側面を階層的に記述される。データベースを読み出した後で必要な情報を選択することで相違するデータベースの組み合わせによる動作を可能にしている。小分類は、実際のハードウェア構造の形になる（単位系などはこの時点で決まる）。

【０１１８】センサ値情報フィールドは、該当するセンサ入力の最大値、最小値、並びに、解像度を記入するフィールドを含んでいる。

【０１１９】センサ識別情報（SensorID）フィールドは、センサ部品並びにセンサが搭載された部位を指定する各フィールドと、部品並びに部位に関する互換性を示す各フラグを含んでいる。

【０１２０】分類並びにセンサ識別情報がデータベース検索時において該当するレコードを選択する決め手となる。

【０１２１】アプリケーションなどのソフトウェアは、実行時に組み合わされているミドルウェアのインフォメーション・データベースを検索して、必要としているインフォメーション識別情報を獲得して、動作時に通知される情報群からインフォメーション識別情報を検索キーにして情報を抽出し、必要な処理を適用する。

【０１２２】図５及び図６には、インフォメーション・データベースのレコードの記入例を示している。

【０１２３】各図において、インフォメーションＩＤが１０１と１０２のレコードは実際には同じセンサを指定している。同様に、２００と２０１のレコードは実際には同じセンサを指定している。１０１に比し、１０２にはより詳細な部位情報が記述されている。また、２０１はＰＳＤが測定している場所とカメラが捕捉している場所との関係を画像連動型として得る点で、２００とは相違する。

【０１２４】アプリケーションは、このインフォメーション・データベースを用いて、使用する入力情報を登録しておくことにより、ミドルウェアから該当するセンサ入力情報の通知を受けることかできる。

【０１２５】図７には、アプリケーション・オブジェクトが使用する情報を登録する様子、並びに、登録された情報がミドルウェア・オブジェクト経由でアプリケーション・オブジェクトに通知される様子を模式的に示している。以下、同図を参照しながら、アプリケーション・オブジェクトが使用情報を登録する仕組み、並びに登録した情報をアプリケーション・オブジェクトが受け取る仕組みについて説明する。

【０１２６】アプリケーション・オブジェクトは、例えば、「なでる」ことに関するセンサ値情報を使用したい場合には、「なでる」という意味ベースのデータベース要求をオブジェクト管理システムに対して発行する（Ｔ１）。

【０１２７】オブジェクト管理システムは、このデータベース要求に応答して、「なでる」という意味ベースの分類を検索キーにして、インフォメーション・データベースを検索する。そして、該当するレコードを見つけ出し、そのインフォメーション識別情報を要求元のアプリケーション・オブジェクトに戻す（Ｔ２）。

【０１２８】アプリケーション・オブジェクトは、この戻り値に満足した場合には、オブジェクト管理システムに対して当該情報を使用する旨の宣言（すなわち登録）を行うとともに（Ｔ３）、ミドルウェア・オブジェクトに対してレディ状態すなわち該当するセンサ値情報の入力がＯＫであることを通知する（Ｔ４）。また、オブジェクト管理システムは、この宣言に応答して、ミドルウェア・オブジェクトに対して登録確認を行う（Ｔ５）。

【０１２９】ミドルウェア・オブジェクトは、登録確認に応答して、インフォメーション・データベースの該当レコードが指定するセンサをハードウェア操作するドライバ・オブジェクトに対して、センサの付勢（すなわちセンサ・オープン）を指示する（Ｔ６）。

【０１３０】センサがオープンされる結果として、ミドルウェア・オブジェクトは、センサ値情報を時々刻々入力することができる（Ｔ７）。ミドルウェア・オブジェクトは、各センサからの入力情報を、登録内容に従ってアプリケーション・オブジェクト毎にまとめて、通知する（Ｔ８）。

【０１３１】センサ値情報（Sensor Information）

アプリケーション・オブジェクトがインフォメーション・データベースを用いて使用情報を登録することによって、所望のセンサ値情報を入力可能な状態になる仕組みは上述した通りである。次いで、アプリケーション・オブジェクトがセンサ値情報を取得する仕組み、並びにセンサ値情報の詳細について説明する。

【０１３２】図８には、アプリケーション・オブジェクトがミドルウェア・オブジェクトに対してセンサ値情報を要求する仕組みを図解している。

【０１３３】アプリケーション・オブジェクトは、インフォメーション・データベースを基に、必要な情報を得るためのインフォメーション識別情報（InformationID）を取得することができる。

【０１３４】アプリケーション・オブジェクトは、このインフォメーション識別情報を用いて、ミドルウェア・オブジェクトに対してデータ読み取り（Ｒｅａｄ）要求を発行する。ミドルウェア・オブジェクトは、このＲｅａｄ要求に応答して、ドライバ・オブジェクト経由で入

力されるセンサ値情報を、アプリケーション・オブジェクトに通知する。

【０１３５】図９には、アプリケーション・オブジェクトがセンサ値情報を取得するための、各オブジェクト間のやり取りをチャート上に模式的に示している。

【０１３６】同図に示すように、アプリケーション・オブジェクトは、センサ値情報が必要となったときに、ミドルウェア・オブジェクトに対してＲｅａｄ要求を発行する。

【０１３７】ミドルウェア・オブジェクトは、ドライバ・オブジェクト経由で、各センサからセンサ値を入力している。アプリケーション・オブジェクトからのＲｅａｄ要求に応答して、要求されたインフォメーション識別情報に該当するセンサ値を、アプリケーション・オブジェクトに戻す。但し、アプリケーション・オブジェクトがレディ（Ready）状態でないときに入力されたセンサ値は、アプリケーション・オブジェクトに転送せず、廃棄する。

【０１３８】図１０には、センサ値情報のデータ構造を模式的に図解している。同図に示すように、センサ値情報は、ヘッダ部と本体部で構成される。

【０１３９】ヘッダ部は、ヘッダ情報（HeaderInfo）とセンサ情報（SensorInfo）で構成される。ヘッダ部には、インフォメーション・データベースに記述できない、動作中に変化するタイプの情報を付加することができる。

【０１４０】一度に複数の情報が送られるので、インフォメーション識別情報で該当するヘッダを検索することができる。検索したヘッダにバリュー情報へのオフセットが書き込まれる。バリューはデータ列で構成され、該当するセンサ値の抽出は、ヘッダ情報並びにセンサ情報から行うことができる。

【０１４１】図１１には、ヘッダ情報のデータ構造を模式的に図解している。同図に示すように、ヘッダ情報は、ヘッダ・サイズ（HeaderSize）と、本体サイズ（BodySize）と、センサ数（NumOfSensor）と、時間（Time）と、ロボット速度（RobotSpeed）と、姿勢識別情報（PostureID）とで構成される。

【０１４２】ロボット速度（RobotSpeed）は、同図に示すように、ｘｙｚ各軸方向の並進速度成分と回転速度成分が書き込まれる。

【０１４３】姿勢識別情報（PostureID）は、共通概念として定義されたものを使用する。姿勢識別情報は、以下のようなデータ構造で定義される。

【０１４４】

【数１】TypeDef PostureID byte

【０１４５】また、図１２には、センサ情報（SensorInfo）のデータ構造を模式的に示している。センサ情報（SensorInfo）には、本体部のフォーマットを記述することができる。

【０１４６】センサ情報（SensorInfo）は、インフォメーション識別情報（Information ID）と、本体部の中から分類で規定された情報にアクセスするための情報を含んでいる。本体部にアクセスするための情報は、本体部のオフセット量（offset）、垂直方向サイズ（vertical）、水平方向サイズ（horizontal）、スキップ数（skip）、ステップ数（step）などで構成される。

【０１４７】otimeは、このセンサ値情報を検出した時刻のことであり、フィルタの遅延時間などの影響を受ける。Limitは、センサ自身が端点にあるので動かすことができないか否かを示すフラグである。

【０１４８】Tmatrixは、センサの状態を表す情報であり、より具体的には、センサ座標を基準座標に変換するための４×４行列である。センサがTmatrixにより基準座標の位置に変換された様子を、図１３に示しておく。

【０１４９】フィードバック・インターフェース

本実施例に係るアプリケーション・インターフェースは、アプリケーションで検出した認識結果をミドルウェアに通知して、認識結果とミドルウェアで可能な行動との関係をアプリケーションに通知するというフィードバック・ループを実現する。フィードバック・ループのためのインターフェース、すなわち「フィードバック・インターフェース」が実装される。

【０１５０】図１４には、フィードバック・インターフェースを用いたフィードバック・ループにより通知されたコマンド情報に従い、アプリケーション・オブジェクトがコマンドを発行する様子を図解している。

【０１５１】ミドルウェア・オブジェクトは、仕様情報の登録内容に基づき、画像データなどのセンサ値情報をアプリケーション・オブジェクトに送信する。

【０１５２】これに対し、アプリケーション・オブジェクトは、センサ値情報に基づいて所定のターゲットを認識処理し、認識結果として目的語形式のターゲット情報（TargetInfo）をミドルウェア・オブジェクトに戻す。

【０１５３】さらに、ミドルウェア・オブジェクトは、ターゲットに対して可能なインタラクションに関するコマンド情報（CommandInfo）をアプリケーション・オブジェクトにフィードバックする。

【０１５４】アプリケーション・オブジェクトは、コマンド情報を基に、選択可能なコマンドを選んで、ミドルウェア・オブジェクトにコマンド発行する。ミドルウェア・オブジェクトは、ドライバ・オブジェクトを介して所定のハードウェア操作を行い、コマンドをロボットの機体上で体現させることができる。

【０１５５】図１５には、フィードバック・ループで使用するターゲット情報（TargetInfo）のデータ構造を模式的に示している。

【０１５６】同図に示すように、ターゲット情報は、情報を検出した時刻を示すTimeと、追跡するターゲットを識別するためのAppIDと、使用したセンサを識別するた

めのSensorIDと、基準座標からセンサ座標への変換マトリックスであるTmatrixと、ターゲットの状態を表すTargetStatusと（Unknown, Lost, NotSupportのいずれかの状態をとる）（図１６を参照参照のこと）、センサ座標系でのターゲットの位置を極座標表示するTargetPosと、ターゲットのサイズを直方体で表すSize3Dと（図１７を参照のこと。図１７中の変数Enable3Dのデータ構造は図１８を参照のこと）で構成される。

【０１５７】図１９には、フィードバック・ループで使用するコマンド情報（CommandInfo）のデータ構造を模式的に示している。

【０１５８】同図に示すように、コマンド情報は、情報を検出した時刻を示すTimeと、追跡するターゲットを識別するためのAppIDと、使用したセンサを識別するためのSensorIDと、ターゲットとの関係を示すStatusと（図２０を参照のこと）、インタラクションのコマンド（すなわち選択可能なコマンド）を示すCategoryと、ターゲットと中心の関係を示すPosition3Dとで構成される。

【０１５９】Ｋｉｃｋ－Ｔｏｕｃｈ系のコマンドは、ロボットの構成によっていろいろ変わる。また、Ｔｒａｃｋｉｎｇなどは固定のコマンドとして扱う。

【０１６０】図１９に示すコマンド情報は、Ｃａｔｅｇｏｒｙによる選択の自由度を持たせるタイプである。この変形例として、ＣｏｍｍａｎｄＩＤ列で表されるＡｃｔｉｏｎＩＤを指定する自由度のないタイプを挙げることができる。

【０１６１】システム・イベント

本実施例に係るアプリケーション・インターフェースは、ロボットの機体上で発生した所定の事象すなわちシステム・イベントを検知すると、これをアプリケーション・オブジェクトに通知する機構を備えている。アプリケーション・オブジェクトは、システム・イベントの通知に応答して、所定のコマンドを実行するようになっている。図２１には、アプリケーション・インターフェースを介したシステム・イベントの処理を図解している。

【０１６２】図２２には、ミドルウェア・オブジェクトがアプリケーション・オブジェクトに対してイベント通知に使用するシステム・イベント（SysEvent）のデータ構造を模式的に示している。

【０１６３】同図に示すように、システム・イベントは、イベント発生時刻を示すTimeと、インタラクションのコマンド（すなわち選択可能なコマンド）を示すCategoryと、アクションの個数を示すNactionと、CommandID列で表されるActionIDとで構成される。

【０１６４】アクションを構成する配列型変数ａｉｄ［ｉ］には、システム・イベントに対応する処理を割り当てることができる。ａｉｄの使用例を図２３並びに図２４に示しておく。

【０１６５】システム・イベントは、センサ値情報とは相違し、ヘッダ情報（HeaderInfo）やセンサ情報（SensorInfo）を含まない。

【０１６６】また、アプリケーション・オブジェクトがレディ（Ready）状態でなくても、システム・イベントを廃棄することなく、蓄積しておく。

【０１６７】シャットダウン－レジューム

本実施例に係るアプリケーション・インターフェースは、ミドルウェア・オブジェクトがシャットダウン（Shutdown）の要因をアプリケーション・オブジェクトに通知する機構と、アプリケーション・オブジェクトがミドルウェア・オブジェクトに対してレジューム（Resume）すなわち次にブートするための条件を設定する機構を備えている。

【０１６８】シャットダウンの要因とレジュームの条件は密接に影響する。例えば、放電時のバッテリの温度が上昇し過ぎた場合に、温度が適温まで降下する時間はハードウェアの放熱性能など機体の特性に依存する。ミドルウェア・オブジェクトは、その際のレジューム条件として、次に起きてもよい時間を推奨レジューム条件としてアプリケーション・オブジェクトに通知する。アプリケーション・オブジェクトは、この推奨レジューム条件とアプリケーション・オブジェクト自身の都合から、最終的なレジューム条件を設定して、シャットダウン時にミドルウェア・オブジェクトに通知する。

【０１６９】図２５には、アプリケーション・インターフェースを介したシャットダウン要因の通知とレジューム条件の設定を行う処理の一例を図解している。

【０１７０】例えば、放電時のバッテリの温度が上昇し過ぎたようなPowerEventがシャットダウン要因として検出された場合、ミドルウェア・オブジェクトは、アプリケーション・オブジェクトにPowerEventを通知する。ミドルウェア・オブジェクトがPowerEventを通知する際に、次に起きてもよい時間をレジューム条件を推奨レジューム条件（RecommendCondition）として、併せて通知する。

【０１７１】これに対し、アプリケーション・オブジェクトは、推奨レジューム条件と、アプリケーション・オブジェクト自身の都合から、最終的なレジューム条件（ResumeCondition）を設定して、シャットダウン時にミドルウェア・オブジェクトに通知する。

【０１７２】図２６には、ミドルウェア・オブジェクトがアプリケーション・オブジェクトに対して通知するシャットダウン要因PowerEventのデータ構造を模式的に示している。

【０１７３】同図に示すように、PowerEventは、当該シャットダウン要因が発生した時刻を示すTimeと、インタラクションのコマンド（すなわち選択可能なコマンド）を示すCategoryと、レジューム条件として推奨される条件を示すRecommendConditionと、アクションの個数を示すNactionと、CommandID列で表されるActionIDとで構成される。

27

【０１７４】また、図２７には、アプリケーション・オブジェクトが次のブート条件を設定するためにミドルウェア・オブジェクトに転送するResumeConditionのデータ構造を模式的に示している。

【０１７５】同図に示すように、ResumeConditionは、イベント通知時のマスクを示すMaskと、シャットダウン時のイベントを示すEventと、次のレジューム時刻を設定するResumeTimeとで構成される。

【０１７６】コマンド・データベース（CommandDB）

本実施例に係るアプリケーション・インターフェースは、アプリケーション・オブジェクトがミドルウェア・オブジェクトを制御するためのコマンド・インターフェースを提供する。このコマンド・インターフェースがサポートするコマンドの中で、アプリケーション・オブジェクトで実行したいコマンドを意味ベースで選択できるようにするために、ミドルウェア・オブジェクトは、コマンド・データベースを用意する。アプリケーション・オブジェクトは、ミドルウェア・オブジェクトが実際のどのようなモータ／エフェクタを作動させるのかを意識することなく、必要な行動を実行させることができる。

【０１７７】図２８には、コマンド・データベースのフォーマット例を示している。同図に示すように、コマンド・データベースの１つのレコードは、コマンド識別情報（CommandID）フィールドと、分類フィールドと、リソース・フィールドと、姿勢情報フィールドと、センサ識別情報（SensorID）フィールドと、無限実行フラグ・フィールドと、世代識別情報（世代ＩＤ）フィールドと、程度識別情報（程度ＩＤ）フィールドとで構成される。

【０１７８】コマンド識別情報（CommandID）は、レコードに一義に割り振られる固有の識別子であり、ミドルウェアが勝手に決めることができる。

【０１７９】分類フィールドは、さらに、互換、大、中、小の各フィールドに細分化される。互換フィールドは、その後に書かれるＩＤの意味が標準又は拡張のＩＤのいずれであるかを示すフラグである。大、中、小の各フィールドには大分類、中分類、小分類がそれぞれ記入される。分類は、意味ベースで情報を選択することを意識して、すなわち情報の持つ意味ベースの側面を階層的に記述される。

【０１８０】リソース・フィールドは、動作（頭，尻尾，足）、音（スピーカは１つ）、特殊（頭ＬＥＤ、尻尾ＬＥＤ）という各管理単位毎のビット・フィールドに細分化される。ビット・フィールドの各ビット値の意味は、ミドルウェアの性能次第で割り当てを変更することができる。

【０１８１】姿勢情報フィールドは、コマンド実行の際の開始姿勢と終了姿勢、及び姿勢変動（開始と終了で姿勢が同一でも姿勢が変化する場合あり）を指定する各フィールドに細分化される。

28

【０１８２】無限実行フラグは、実行の終了がないことを示すフラグである。

【０１８３】センサ識別情報（SensorID）フィールドは、コマンドを実行する際に利用するセンサ部品並びにセンサが搭載された部位を指定する各フィールドと、部品並びに部位に関する互換性を示す各フラグを含んでいる。

【０１８４】世代識別情報（世代ＩＤ）フィールドは、大人、子供、幼児など各世代毎に利用可能なコマンドか否かを対応ビット位置のビット値で表現するフィールドである。例えば、０１１は、子供と幼児に共通で利用できることを示す。

【０１８５】程度識別情報（程度ＩＤ）フィールドは、コマンドの見た目の程度（例えば運動量など）を表す指標値を記入するフィールドである。

【０１８６】最大世代数や最大程度などは、図示のレコードとは別に定義することになる。

【０１８７】コマンド・データベースは、図示しないヘッダ部を含んでおり、それが合致していれば拡張の意味を知っていて利用することができる。

【０１８８】アプリケーション・オブジェクトなどのソフトウェアは、実行時に組み合わされているミドルウェア・オブジェクトのコマンド・データベースを検索して、実行したい行動を、現在の本能や感情、あるいはその他のあらかじめ定義されているシナリオに応じて選択することができる。選択した行動に割り当てられているコマンド識別情報をミドルウェア・オブジェクトに渡して、該当する今度を実行させることで、意図した行動をロボットの機体上で実現することができる。

【０１８９】アプリケーション・オブジェクトは、リソースが重ならないようにコマンドを選択したりすることで、ロボットの機体上での表現力を豊かにすることができる。また、ロボットが暇な状態では、無限実行フラグが設定されているコマンドを選択して、何らかの動作を継続させるようにすることができる。

【０１９０】図２９及び図３０には、コマンド・データベースのレコードの記入例を示している。

【０１９１】コマンドＩＤが１００，１０１，１０２のレコードは、すべて「うれしい」を表現するコマンドである。アプリケーション・オブジェクトは、ロボットの機体上での状況（例えば、疲労度合いや世代など）に応じて、いずれかのコマンドを選択して、どのレコードの「うれしい」を使用するかを指定することができる。

【０１９２】コマンドＩＤが２００と２０１のレコードは、ともにある対象物のキックを指示するコマンドである。その行動がどんな動作、音、ＬＥＤなのかを知っていて、確実にその行動を実行したい場合に利用することができる。

【０１９３】コマンドーステータス

本実施例に係るアプリケーション・インターフェース

は、アプリケーション・オブジェクトがミドルウェア・オブジェクトを制御するためのコマンド・インターフェースを提供する。アプリケーション・オブジェクトのコマンド発行に対して、ミドルウェア・オブジェクトはドライバ・オブジェクトのようなエージェントを介してロボットの機体上でコマンドを実行するとともに、アプリケーション・オブジェクトに対してはステータスを返すようになっている。

【０１９４】図３１には、アプリケーション・オブジェクトとミドルウェア・オブジェクト間でのコマンド発行とステータス通知を行う仕組みを図解している。

【０１９５】アプリケーション・オブジェクトは、コマンド・データベースを基に、現在の本能や感情、あるいはその他のあらかじめ定義されているシナリオに適応した行動を実現するためのコマンドのコマンド識別情報（CommandID）を取得することができる。

【０１９６】アプリケーション・オブジェクトは、ミドルウェア・オブジェクトがレディー状態のときに、コマンドを発行する。これに対し、ミドルウェア・オブジェクトはドライバ・オブジェクトのようなエージェントを介してロボットの機体上でコマンドを実行するとともに、アプリケーション・オブジェクトに対してはステータスを返す。

【０１９７】図３２には、アプリケーション・オブジェクトが発行したコマンドを実行するための、各オブジェクト間におけるやり取りをタイム・チャート上に模式的に示している。

【０１９８】アプリケーション・オブジェクトがミドルウェア・オブジェクトに対してコマンドを発行する。これに対し、ミドルウェア・オブジェクトは、リソースと姿勢をロック状態にして他のコマンドによる使用を不許可状態にするとともに、ステータスとしてリソース及び姿勢のロック情報をアプリケーション・オブジェクトに返す。その後、ミドルウェア・オブジェクトは、レディ状態である旨をアプリケーション・オブジェクトに通知する。

【０１９９】ミドルウェア・オブジェクトは、ドライバ・オブジェクトのようなエージェントに対してコマンドの実行を要求する。また、ドライバ・オブジェクトからは、入力されたセンサ情報などが返される。

【０２００】コマンドの実行が完了すると、ミドルウェア・オブジェクトは、次のコマンド実行のために、ロック状態を解除し、リソースを解放する。また、アプリケーション・オブジェクトに対しては、リソース解放情報と機体の姿勢情報を含むステータスを返す。

【０２０１】アプリケーション・オブジェクトがコマンドをＮ回あるいは無限回連続して発行する場合も、基本的には、１回のコマンド発行における手順と同様に、ミドルウェア・オブジェクトはステータスを返す。最終コマンドをドライバ・オブジェクトが実行する前に、リソース解放情報と姿勢情報をアプリケーション・オブジェクトに供給することによって、アプリケーション・オブジェクトはリソースの状況を把握することができ、次に選択可能なコマンドの範囲を把握することができる。

【０２０２】図３３には、アプリケーション・オブジェクトが発行するコマンドのデータ構造を模式的に図解している。

【０２０３】同図に示すように、コマンドは、コマンドＩＤを指定するMW-IDと、サブジェクト識別情報（SubjectID）を指定するTargetIDと、センサ座標系でのターゲットの位置を極座標表示するTargetPosとで構成される（但し、TargetInfoのTargetPosがセンサ中心であるのに対し、ここで言うTargetPosの場合、与えた座標はセンサ部位が本体に固定した座標である）。

【０２０４】MW-IDは、使用するコマンドの識別情報（CommandID）を指定するためのフィールドである。個々で指定可能なコマンドは、「通常」と、「緊急停止」と、「停止」の３種類がある。

【０２０５】通常のコマンドは上書き実行される。音声やＬＥＤに関するコマンドは緊急停止の場合と同じである。モーションは停止の場合と同じである。緊急停止は、安定状態や姿勢の保証がない。停止は、安定した状態や姿勢で停止し、基本は通常終了を待つのと等価である。

【０２０６】図３４には、TargetIDのデータ構造を模式的に図解している。SubjectIDに含まれるOIDが０×０であれば、TargetPosが有効となる。

【０２０７】図３５には、ミドルウェア・オブジェクトがアプリケーション・オブジェクトに返すステータスのデータ構造を模式的に図解している。同図に示すように、ステータスは、コマンドＩＤを指定するMW-IDと、コマンドの状態を示すCommandStatusと、アクションの状態を示すActionStatusと、姿勢の識別情報PostureIDと、リソース情報ResourceInfoとで構成される。

【０２０８】コマンド状態CommandStatusは、'Complete'、'Not Support'、'Incomplete'の３状態が定義されている。実行すべきコマンドがなかった場合や緊急停止した場合にはIncompleteとなる。

【０２０９】アクション状態ActionStatusは、'Success'（実行の成功を検出した場合）、'Fail'（実行の失敗を検出した場合）、'Unknown'（実行の判定機能がない場合）の３状態が定義されている。

【０２１０】リソース情報ResourceInfoは、ミドルウェアで使用するリソース情報を記述する。図３６には、ResourceInfoのデータ構造を模式的に示している。同図に示すように、ResourceInfoは、モーション、音声（スピーカ）、ＬＥＤなどの各リソースがビットマップ形式で記述される。各ビットとリソースの対応は自由に定義することができる。

【０２１１】姿勢の識別情報PostureIDは、現在の姿勢

を記述する。

【０２１２】［追補］以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。

【０２１３】本発明の要旨は、必ずしも「ロボット」と称される製品には限定されない。すなわち、電気的若しくは磁気的な作用を用いて人間の動作に似せた運動を行う機械装置であるならば、例えば玩具等のような他の産業分野に属する製品であっても、同様に本発明を適用することができる。

【０２１４】要するに、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０２１５】

【発明の効果】以上詳記したように、本発明によれば、脚式歩行型など多関節型のロボットをソフトウェア・プログラムを用いて制御する、優れたロボット制御システム及びロボット制御方法を提供することができる。

【０２１６】また、本発明によれば、脚部や頭部など各動作ユニットの着脱・交換などに伴ってハードウェア構成が大幅に変更する可能性がある多関節型ロボットをソフトウェア・プログラムを用いて制御することができる、優れたロボット制御システム及びロボット制御方法を提供することができる。

【０２１７】また、本発明によれば、ハードウェア構成に対して依存性の高いソフトウェア層とハードウェア構成に非依存のソフトウェア層の組み合わせからなるソフトウェア・プログラムを用いて多関節型ロボットを制御することができる、優れたロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースを提供することができる。

【０２１８】また、本発明によれば、ミドルウェアのようなハードウェア依存のソフトウェア層と、アプリケーションなどのハードウェア非依存のソフトウェア層との組み合わせを動的に変更して多関節型ロボットを制御することができる、優れたロボット制御システム及びロボット制御方法、並びに各ソフトウェア層間のプログラム・インターフェースを提供することができる。

【０２１９】本発明によれば、ロボットのハードウェア構成に依存するミドルウェア層と、ハードウェア構成に依存しないアプリケーション層の間でのプログラム・インターフェースを確立することによって、ロボット上に導入するミドルウェアとアプリケーションの組み合わせを変更しても、常に正常な動作を保証することができる。

【０２２０】また、本発明によれば、アプリケーションとミドルウェアとの間で、意味的に動作を行うためのインターフェースとデータベースを用意することによって、各アプリケーションは、ミドルウェアを介して適当な入力データを取得することができるとともに、ハードウェア操作用の適切なコマンドをミドルウェアに発行することができる。

【図面の簡単な説明】

【図１】本発明の実施に供されるロボットのハードウェア構成を模式的に示した図である。

【図２】ＣＰＣコンポーネントの着脱・交換により駆動系サブシステム５０の構成を変更した例を示した図である。

【図３】ロボット制御用のソフトウェアの構成を説明するための図である。

【図４】インフォメーション・データベースのフォーマット例を示した図である。

【図５】インフォメーション・データベースのレコードの記入例を示した図である。

【図６】インフォメーション・データベースのレコードの記入例を示した図である。

【図７】アプリケーションが使用する情報を登録する様子、並びに、登録された情報がミドルウェア経由で通知される様子を模式的に示した図である。

【図８】アプリケーション・オブジェクトがミドルウェア・オブジェクトに対してセンサ値情報を要求する仕組みを示した図である。

【図９】アプリケーション・オブジェクトがセンサ値情報を取得するための、各オブジェクト間のやり取りを模式的に示したチャートである。

【図１０】センサ値情報のデータ構造を模式的に示した図である。

【図１１】ヘッダ情報のデータ構造を模式的に示した図である。

【図１２】センサ情報（SensorInfo）のデータ構造を模式的に示した図である。

【図１３】センサがTmatrixにより基準座標の位置に変換された様子を示した図である。

【図１４】フィードバック・インターフェースを用いたフィードバック・ループによって通知されたコマンド情報に従い、アプリケーション・オブジェクトがコマンドを発行する様子を示した図である。

【図１５】フィードバック・ループで使用するターゲット情報（TargetInfo）のデータ構造を模式的に示した図である。

【図１６】ターゲットの状態を表すTargetStatusのデータ構造を模式的に示した図である

【図１７】ターゲットのサイズを直方体で表すSize3Dのデータ構造を模式的に示した図である。

【図１８】図１７中の変数Enable3Dのデータ構造を模式的に示した図である。

【図１９】フィードバック・ループで使用するコマンド情報（CommandInfo）のデータ構造を模式的に示した図

33

である。

【図２０】ターゲットとの関係を表すStatusのデータ構造を模式的に示した図である。

【図２１】アプリケーション・インターフェースを介したシステム・イベントの処理を模式的に示した図である。

【図２２】システム・イベント（SysEvent）のデータ構造を模式的に示した図である。

【図２３】ａｉｄの使用例を示した図である。

【図２４】ａｉｄの使用例を示した図である。

【図２５】アプリケーション・インターフェースを介したシャットダウン要因の通知とレジューム条件の設定を行う処理の一例を示した図である。

【図２６】ミドルウェア・オブジェクトがアプリケーション・オブジェクトに対して通知するシャットダウン要因PowerEventのデータ構造を模式的に示した図である。

【図２７】アプリケーション・オブジェクトが次のブート条件を設定するためにミドルウェア・オブジェクトに転送するResumeConditionのデータ構造を模式的に示した図である。

【図２８】コマンド・データベースのフォーマット例を示した図である。

【図２９】コマンド・データベースのレコードの記入例を示した図である。

34

【図３０】コマンド・データベースのレコードの記入例を示した図である。

【図３１】アプリケーション・オブジェクトとミドルウェア・オブジェクト間でのコマンド発行とステータス通知を行う仕組みを示した図である。

【図３２】アプリケーション・オブジェクトが発行したコマンドを実行するための、各オブジェクト間のやり取りをチャート上に模式的に示した図である。

【図３３】アプリケーション・オブジェクトが発行するコマンドのデータ構造を模式的に示した図である。

【図３４】TargetIDのデータ構造を模式的に示した図である。

【図３５】ミドルウェア・オブジェクトがアプリケーション・オブジェクトに返すステータスのデータ構造を模式的に示した図である。

【図３６】リソース情報（ResouceInfo）のデータ構造を模式的に示した図である。

【符号の説明】

１０…制御系サブシステム

１１…ＣＰＵ

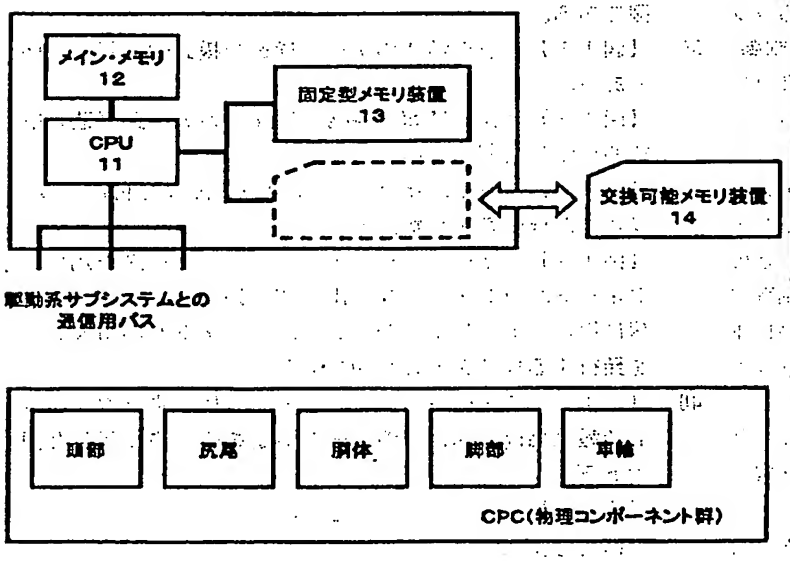
１２…メイン・メモリ

１３…固定型メモリ装置

１４…交換可能メモリ装置

５０…駆動系サブシステム

【図１】



【図１６】

```
enum TargetStatus{
        Found;
        Lost;
        Unknown;
}
```

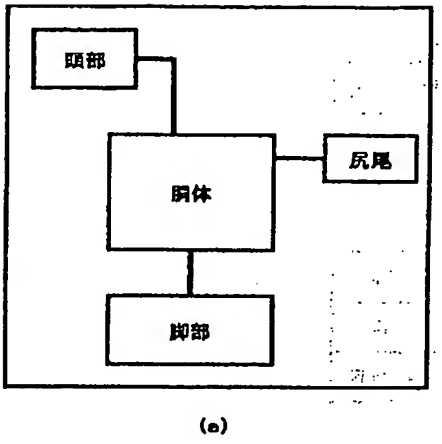
【図１７】

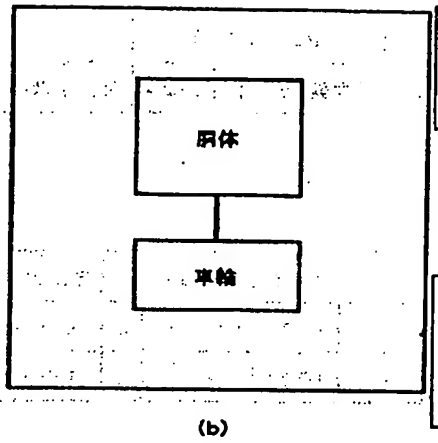
```
struct Size3D{
        Enable3D    enable;
        int         width;
        int         hight;
        int         depth;
}
```

【図１８】

```
bitmap Enable3D{
        bit   FixZero[5];
        bit   First;
        bit   Second;
        bit   3-rd;
}
```

【図２】





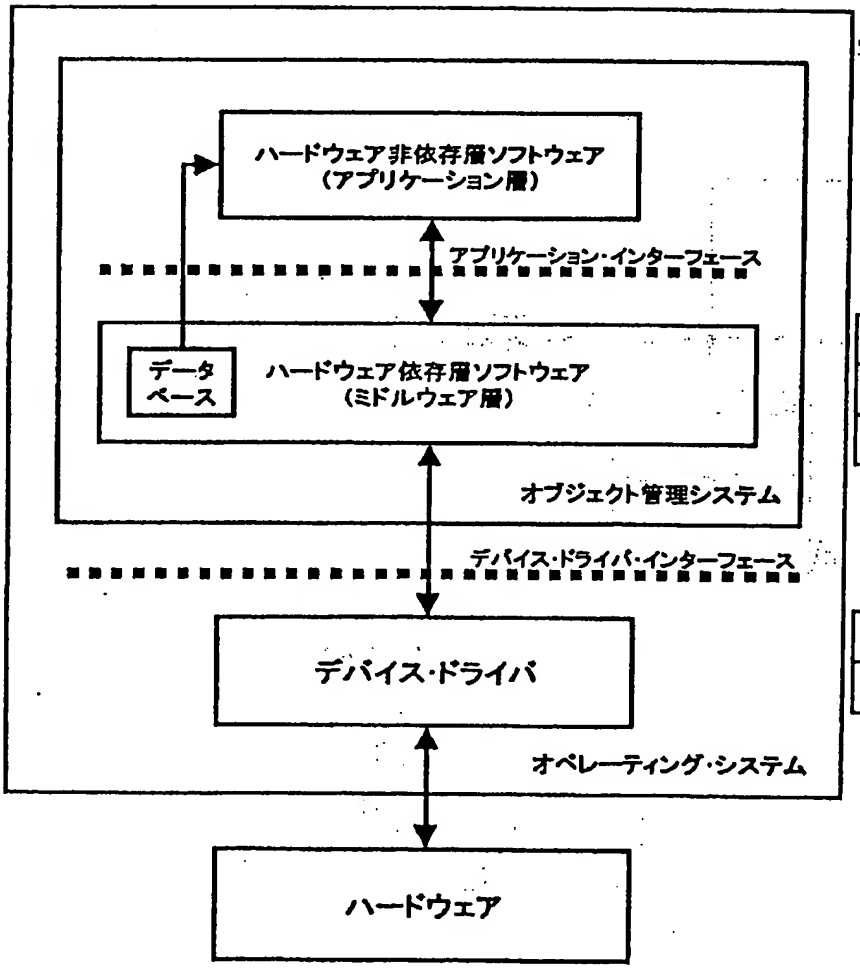
【図２０】

```
enum Status{
        Near;
        Just;
        Far;
}
```

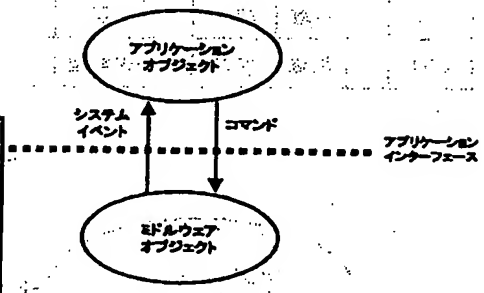
【図２２】

```
struct SysEvent{
        Time        otime;
        Category    category;
        Naction     num;
        ActionID    aid[];
}
```

【図３】



【図２１】



【図２３】

aid[0]	Emergency Stop
aid[1]	Gain Off
aid[2]	割り当て動作

【図２４】

aid[0]	Emergency Stop
aid[1]	割り当て動作

【図４】

InformationID	分類				センサ値情報			SensorID			
	互換	大	中	小	最大値	最小値	解像度	互換	部品	互換	部位

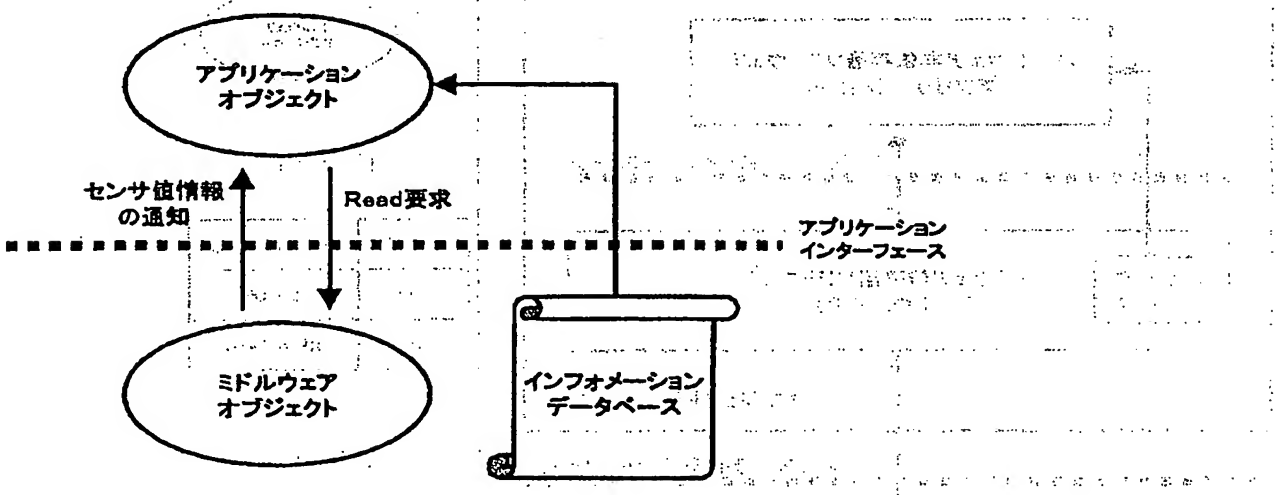
【図５】

100	0	接触	なでる	接触圧力型	0	253	1(Pal)	0	感圧センサ	0	頭
101	0	接触	さわる	接触時間型	0	1024	8(msec)	0	スイッチ	0	足
102	0	接触	さわる	接触時間型	0	1024	8(msec)	0	スイッチ	1	右前足

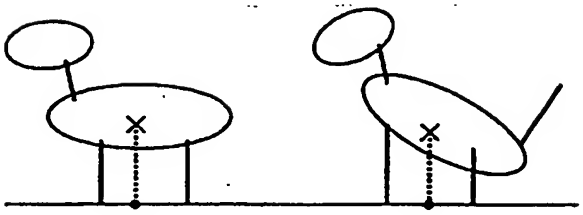
【図６】

200	0	距離	前方	距離型	0	100	10(cm)	0	PSD	1	頭
201	0	距離	前方	画像連動型	0	100	10(cm)	0	PSD	1	頭

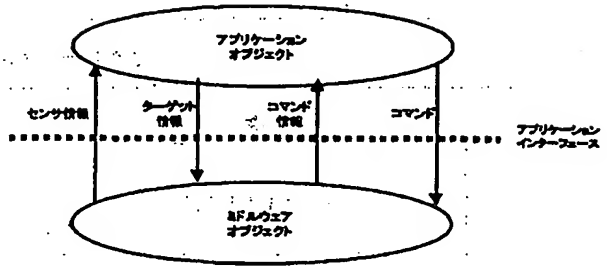
【図８】



【図１３】



【図１４】



【図７】

【図２８】

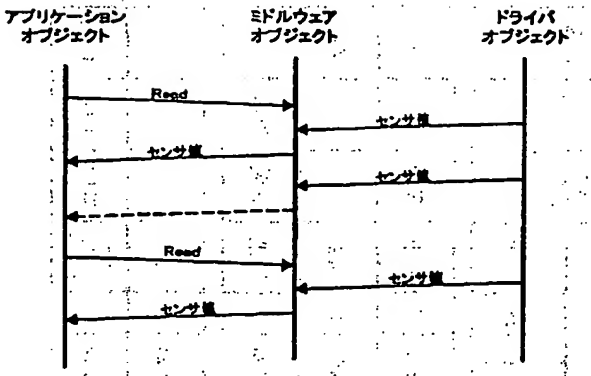
CommandID		分類			リソース			姿勢情報			SensorID				無限実行Flag	世代ID	程度ID
		互換	大	中	小	動作	音	特殊	開始	終了	変動	互換	部品	互換	部位			
100	0	感情表現	うれしい		うれしい	111	1	11	立ち	立ち	立ち	あり	0	無効	0	無効	有限	011	80
101	0	感情表現	うれしい		うれしい	001	0	00	座り	立ち	立ち	なし	0	無効	0	無効	有限	110	40
102	0	感情表現	うれしい		うれしい	000	1	11	全姿勢	全姿勢	全姿勢	なし	0	無効	0	無効	有限	001	10

【図２９】

CommandID		分類			リソース			姿勢情報			SensorID				無限実行Flag	世代ID	程度ID
		互換	大	中	小	動作	音	特殊	開始	終了	変動	互換	部品	互換	部位			
200	0	物への接触	前方		キック	001	1	01	立ち	立ち	立ち	なし	0	無効	0	足	有限	110	20
201	1	物への接触	前方		キック	001	1	01	立ち	立ち	立ち	なし	0	無効	1	左後ろ足	有限	110	100

【図３０】

【図９】



【図１０】

```
struct SensorInformation{
    HeaderInfo      hinfo;
    SensorInfo      info[1];
}
```

ヘッダ部

```
struct SensorVector{
        Value       value[1];
}
```

本体部

【図１１】

```
struct HeaderInfo{
    HeaderSize      hsize;
    BodySize        bsize;
    NumOfSensor     nsensor;
    Time            otime;
    RobotSpeed      speed;
    PostureID       pid;
}
```

```
struct RobotSpeed{
    Vvalue  Vx;  //  並進速度成分x
    Vvalue  Vy;  //  並進速度成分y
    Vvalue  Vz;  //  並進速度成分z
    Vvalue  Wx;  //  回転速度成分x
    Vvalue  Wy;  //  回転速度成分y
    Vvalue  Wz;  //  回転速度成分z
}
```

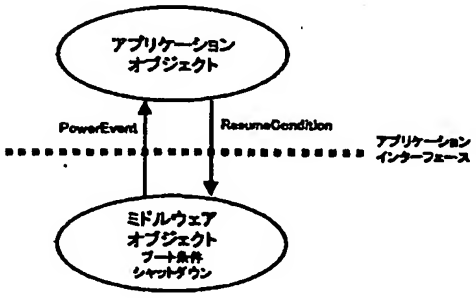
【図１５】

```
struct TargetInfo{
    Time        otime;   // 情報を検出した時刻
    AppID       target;  // ターゲット
    SensorID    sid;     // 使用したセンサ
    Tmatrix     matrix;  // 基準座標からセンサ座標への変換マトリックス
    Targetstatus status; // Unknown - Lost - NotSupport
    TargetPos   tdata;   // センサ座標でのターゲットの位置(視座標表示)
    Size3D      size;    // ターゲットのサイズ(直方体)
}
```

【図１９】

```
struct CommandInfo{
    Time        otime;     // 情報を検出した時刻
    AppID       target;    // ターゲット
    SensorID    sid;       // 使用したセンサ
    Status      status;    // ターゲットとの関係
    Category    category;  // インタラクションのコマンド(KickTouch)
    Position3D  position;  // ターゲットと中心の関係
}
```

【図２５】



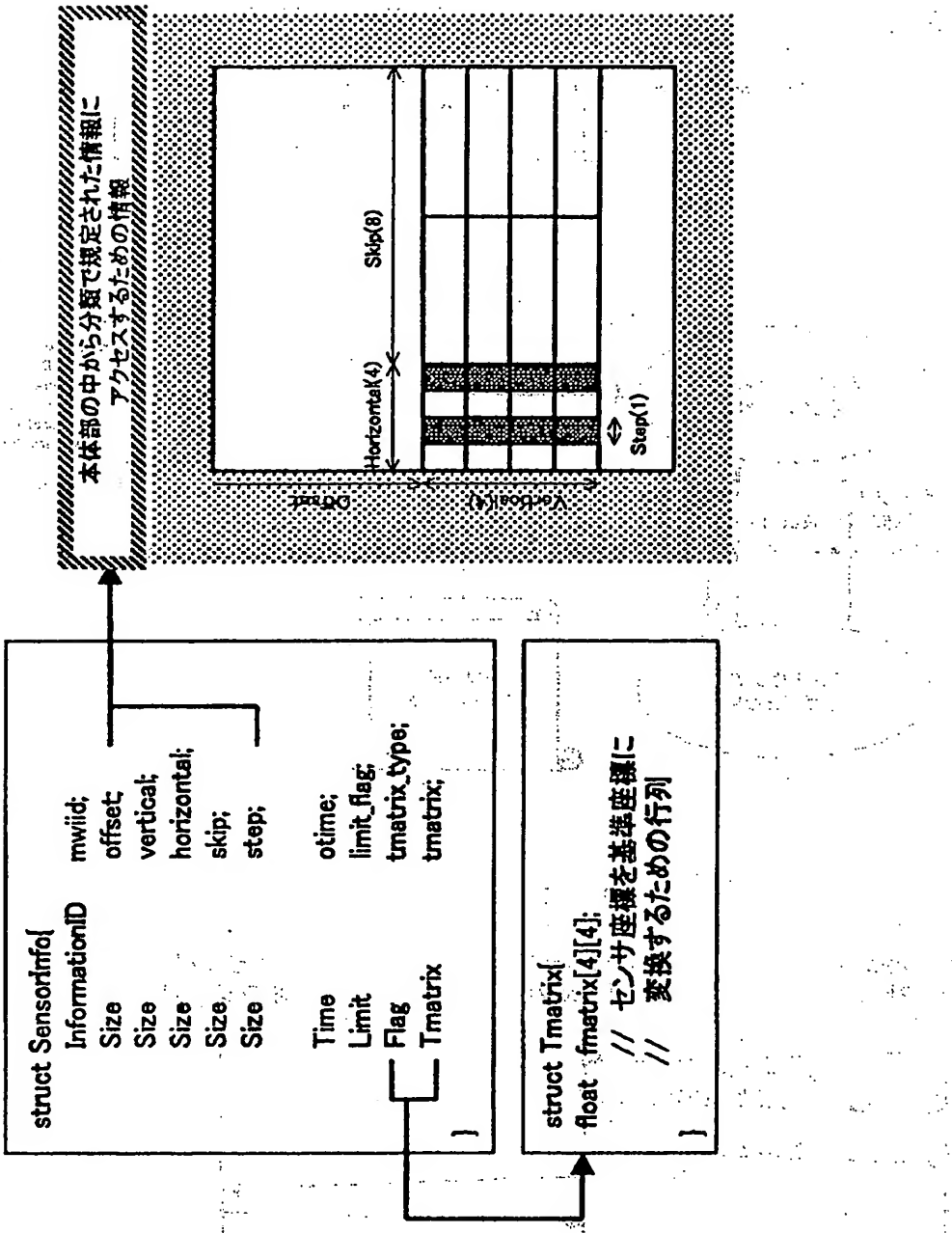
【図３３】

```
struct Command{
    MW-ID       CommandID;
    TargetID    subjectID;
    TargetPos   tdata;
}
```

【図３４】

```
struct TargetID{
    SubjectID   SubjectID;
    AppID       appid;
}
```

【図１２】


本体部の中から分類で規定された情報に
アクセスするための情報
Skip(8)
Horizontal(4)
Step(1)
struct SensorInfo{
InformationID mwiid;
Size offset;
Size vertical;
Size horizontal;
Size skip;
Size step;
Time otime;
Limit limit_flag;
Flag tmatrix_type;
Tmatrix tmatrix;
}
struct Tmatrix{
float fmatrix[4][4];
// センサ座標を基準座標に
// 変換するための行列
}

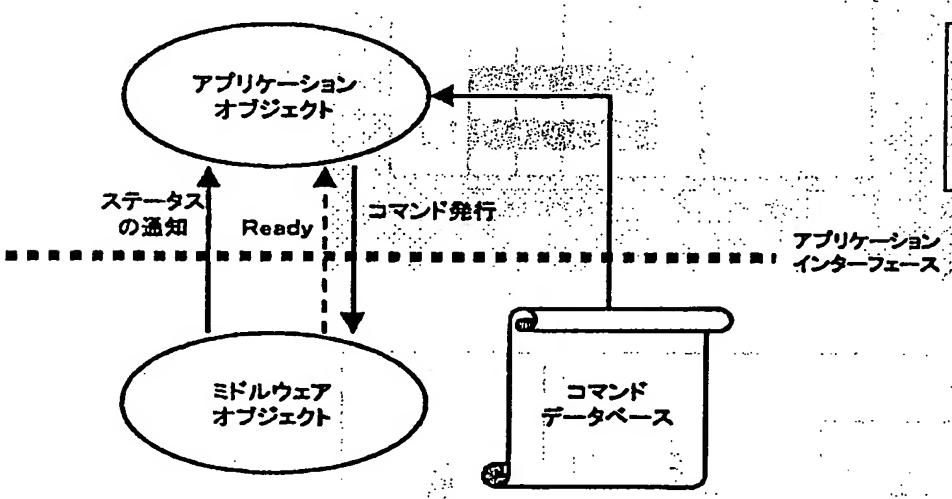
【図２６】

```
struct PowerEvent{
        Time            otime;
        Category        category;
        ResumeCondition RecommendCondition;
        Naction         num;
        ActionID        aid[];
}
```

【図２７】

```
struct ResumeCondition{
        Mask            mask;
        Event           event;
        ResumeTime      time;
}
```

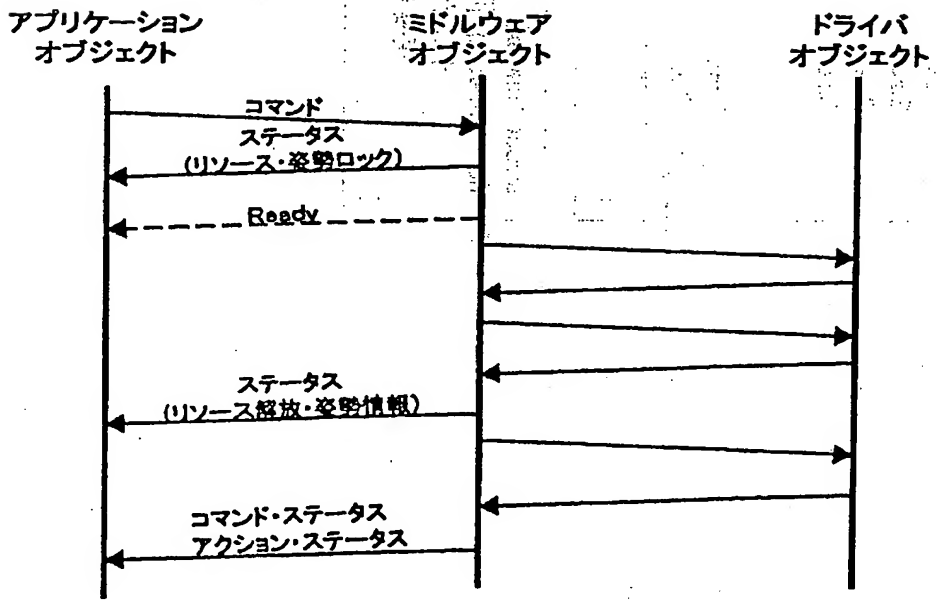
【図３１】



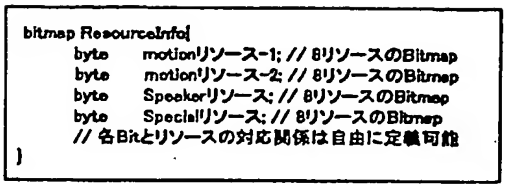
【図３５】

```
struct Status{
        MW-ID           CommandID;
        CommandStatus   Cstatus;
        ActionStatus    Astatus;
        PostureID       posid;
        ResourceInfo    resid;
}
```

【図３２】



【図３６】



フロントページの続き

(72)発明者　細沼　直泰
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　高木　剛
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　藤田　雅博
東京都品川区北品川６丁目７番35号　ソニー株式会社内

Ｆターム(参考)　3C007 AS36 CS08 MT00 WA04 WA14 WB11